Mechanism and Machine Theory 205 (2025) 105885



Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt



Research paper

NURBS curve interpolation strategy for smooth motion of industrial robots

Yonghao Guo, Wentie Niu *, Hongda Liu, Zengao Zhang, Hao Zheng

Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China

A R T I C L E I N F O

Keywords:
NURBS interpolation
Industrial robots
Dynamics constraint
Interpolation output feedrate
Contour error
Roughness

A B S T R A C T

Smooth motion is crucial for industrial robots to efficiently execute accurate path tracking tasks. This paper proposes a NURBS curve interpolation strategy for smooth motion of industrial robots to reduce roughness and contour error. The strategy ensures smooth motion through two stages: feedrate planning and interpolation point parameter calculation. During the feedrate planning stage, kinematics and dynamics constraints, including torque and torque change rate, are considered in the parameter domain. Round-off error is considered, and an S-curve feedrate planning approach is employed to ensure the planned feedrate is smooth after transitioning from the parameter domain to the time domain. In the interpolation point parameter calculation stage, the displacement guidance curve is generated and updated based on the current situation. Interpolation point iteration compensation is conducted to ensure the interpolation output feedrate is smooth. Simulations and experiments are conducted to validate the effectiveness of the proposed strategy. The simulation results indicate that the proposed strategy effectively smooths the interpolation output feedrate while maintaining efficiency. The experimental results show that the strategy effectively reduces roughness and contour error.

1. Introduction

The field of manufacturing has experienced a notable increase in industrial robot usage in recent years. This change is because of the rapid advancements in automation technology and computer science [1]. Path tracking tasks, such as grinding, deburring, milling, and drilling [2], require high efficiency in robot processing and high precision in end-effector contour. Currently, the commonly used input format in the CNC field is G-code, which contains a series of straight lines and arcs. When machining curves and surfaces, it is necessary to approximate the curves as multiple segments of straight lines or arcs, and then perform line or circle interpolation. However, this approach presents the following challenges during machining: First, frequent changes in line segment direction and the frequent acceleration and deceleration result in poor surface finish and low accuracy, thus requiring a reduction in feedrate to solve these issues. Second, the generation of large program files increases the data transmission burden between CAD/CAM applications and CNC systems [3,4]. In contrast, NURBS has a superior ability to control local parts and express shapes, and an interpolation strategy based on NURBS can enhance fabrication precision, ensure machining efficiency, and facilitate a smooth acceleration/deceleration process [5]. Therefore, there is an increasing interest in studying robot interpolation strategies that are based on NURBS curves for path tracking tasks.

The NURBS curve interpolation strategy includes two aspects: feedrate planning and interpolation point parameter calculation.

* Corresponding author.
E-mail address: niuwentie@tju.edu.cn (W. Niu).

https://doi.org/10.1016/j.mechmachtheory.2024.105885
Received 7 August 2024; Received in revised form 20 November 2024; Accepted 7 December 2024
Available online 13 December 2024

Regarding feedrate planning in the field of robotics, research usually focuses on energy optimization [6,7] or time optimization. Time optimization feedrate planning is a typical requirement to increase the productivity of a robotic system. It involves finding the quickest method to track a predetermined path within the workspace of a robotic system. This must be done while adhering to physical constraints, which include maximum torque constraints as well as maximum velocity, acceleration, and jerk [8]. There are various approaches for solving the time-optimal feedrate planning problem. These approaches can be basically divided into three types [9,10]: indirect approach, direct transcription approach, and dynamic programming approach. The indirect approach, also known as the numerical integration approach [11–16], is based on Pontryagin's principle. The maximum or minimum pseudo acceleration is continuously integrated on the phase plane to solve the robot's optimal feedrate curve. The direct transcription approach takes advantage of the non-linear change of variables. Thus, the time-optimal problem is transformed into a certain form of convex optimization [8–10,17–20]. Then, the approach selects the corresponding convex optimization solver to solve it. The dynamic programming approach [21–24] first transforms the phase plane into many regular discrete grids. Then, the approach is used to find the optimal solution.

However, the aforementioned feedrate planning methods for robots are usually conducted in the parameter domain. The planned feedrate needs to be converted from the parameter domain to the time domain. Because the subsequent interpolation point parameter calculation is periodically executed in the time domain. Although these methods can consider constraints such as the maximum torque of each motor, the robot's end-effector acceleration, and the robot's end-effector jerk in the parameter domain. After transitioning to the time domain, the planned feedrate remains under-smoothed. The smooth planned feedrate can avoid the sudden change in the robot's end-effector load, reduce the vibration, and improve the position control accuracy, which is an essential goal of robot trajectory planning [1,25–27]. Moreover, after converting to the time domain, there is a problem that the planned time cannot be strictly divided by the interpolation period, resulting in round-off errors. Although the round-off error is small, it can affect machining accuracy and smoothness of motion and needs to be eliminated during the feedrate planning stage [28–30]. The aforementioned robot feedrate planning method, when directly transformed into the time domain, still exhibits issues with unsmoothness and round-off error. Therefore, a robot smooth feedrate planning method from the parameter domain to the time domain is required to ensure that the planned feedrate is smooth in the time domain.

Regarding interpolation point parameter calculation, feedrate fluctuation and computational time are often considered key indicators of an efficient algorithm [3]. The quality of the products machined by the NURBS interpolation strategy is significantly affected by feedrate fluctuation [31]. Many strategies have been proposed to reduce feedrate fluctuation, mainly divided into three types: approximation methods, predictor-corrector methods, and fitting methods. The approximation method uses a Taylor expansion based on time to approximate the NURBS curve [32]. Approximation methods based on Taylor expansion are widely used during the interpolation procedure. Typically, a first-order Taylor expansion suffices to meet the requirements [33]. If higher precision is needed, a second-order Taylor expansion [34], the Adams-Bashforth method [5] or the Runge-Kutta method [35] can be used instead of the first-order Taylor expansion method. Since these methods introduce truncation errors, feedrate fluctuations are unavoidable. Although higher-order approximations can increase accuracy, they consume computational resources and reduce real-time performance. The predictor-corrector method [4,36–39] maintains feedrate fluctuations within a specified tolerance range by iteratively adjusting parameter compensations. Because of the introduction of an iterative process, this method is not suitable for high-speed real-time applications. The fitting method can also be referred to as the remapping method. This method can pre-calculate the relationship between parameters and arc length offline based on Hermite cubic spline [40], cubic polynomial [3], quintic spline [41–43], seventh-order polynomial [33,44], shorten real-time calculation time and improve parameter calculation accuracy.

However, most of the literature on interpolation point parameter calculation holds that the reason for feedrate fluctuation is due to the lack of an analytical solution for the relationship between the arc length of the NURBS curve and its parameters. These methods aim to reduce feedrate fluctuation by minimizing the deviation between the actual interpolation output step length and the ideal step length. While these interpolation methods help reduce feedrate fluctuations to some extent, they overlook a key factor when applied to robots: the nonlinear kinematic relationship between the robot's joint and its end-effector. This nonlinearity causes a deviation between the actual interpolation output path and the ideal path, making it impossible for the previous methods to accurately calculate the actual interpolation step length. As a result, the interpolation output feedrate inconsistent with the planned feedrate, and thus, the previous method cannot fully solve this issue. If this issue is not further addressed, it will impact machining precision, leading to reduced surface quality and increased contour errors. Therefore, an interpolation method designed for robots is required to ensure smooth interpolation output feedrate.

The interpolation points generated by the interpolation strategy are directly used as the reference input for each joint of the robot, guiding the operation of the robot. This study proposes that the roughness and contour error can be reduced by smoothing the interpolation output feedrate. Based on the above problems, the NURBS curve interpolation strategy for smooth interpolation output feedrate (ISSIOF) is proposed. The major contributions of the article are as follows:

(1) In the feedrate planning stage, the torque and torque change rate constraints are considered in the parameter domain. The round-off error is considered, and an S-curve feedrate planning approach is employed to ensure that the planned feedrate is smooth after transitioning from the parameter domain to the time domain.
(2) In the interpolation point parameter calculation stage, a displacement guidance curve is generated and dynamically updated. Meanwhile, the interpolation point iteration compensation is employed to make sure that the interpolation output feedrate is smooth.
(3) The simulation and experimental results illustrate that the proposed strategy can reduce the roughness and contour error by smoothing the interpolation output feedrate.

The remainder of the article is organized as follows: The main architecture of the proposed ISSIOF strategy is described in Section 2. The first part of the strategy, the smooth feedrate planning from parameter domain to time domain is introduced in Section 3. The second part, the interpolation point parameters calculation with smooth interpolation output feedrate, is presented in Section 4. In Section 5, relevant simulations and experiments are conducted. The conclusions are summarized in Section 6.

2. The architecture of the proposed strategy

The main architecture of the proposed ISSIOF strategy is depicted in Fig. 1. The proposed strategy is divided into four stages: data input and preprocessing, smooth feedrate planning from parameter domain to time domain, interpolation point parameters calculation with smooth interpolation output feedrate, and data output.

In the first stage, NURBS information, feedrate commands, and interpolation period are entered. The algorithm described in [3] is utilized for the preprocessing of the entered NURBS curve, laying the foundation for subsequent feedrate planning.

At the smooth feedrate planning from parameter domain to time domain stage, feedrate planning considering dynamics constraints is conducted in the parameter domain. The feedrate planning considers both dynamics constraints and kinematics constraints. The dynamics constraints include joint torque and joint torque change rate constraints. The kinematics constraints include joint velocity and end-effector velocity constraints. Based on the planned feedrate, the feedrate profile is divided into three parts: acceleration, constant feedrate, and deceleration regions. In each acceleration and deceleration region, the minimum acceleration and jerk within the region are selected as the conservative maximum acceleration and maximum jerk. S-curve feedrate planning based on region expansion is conducted in the time domain to smooth the planned feedrate. The adjacent acceleration region, constant feedrate region, and deceleration region are combined into one interval, and the time of each interval is calculated to determine whether it can be strictly divided by the interpolation period. If it cannot be strictly divided, it is rounded up. And the additional time required for rounding is evenly distributed to the acceleration and deceleration regions of this interval. In the acceleration and deceleration regions, the maximum acceleration is reduced through the bisection method to meet the additional time required for rounding. On the left side of Fig. 1, there are two graphs corresponding to the feedrate planning stage. One graph shows the relationship between parameter and torque, where the red curve represents the method that considers the constraint of torque change rate, and the blue curve represents

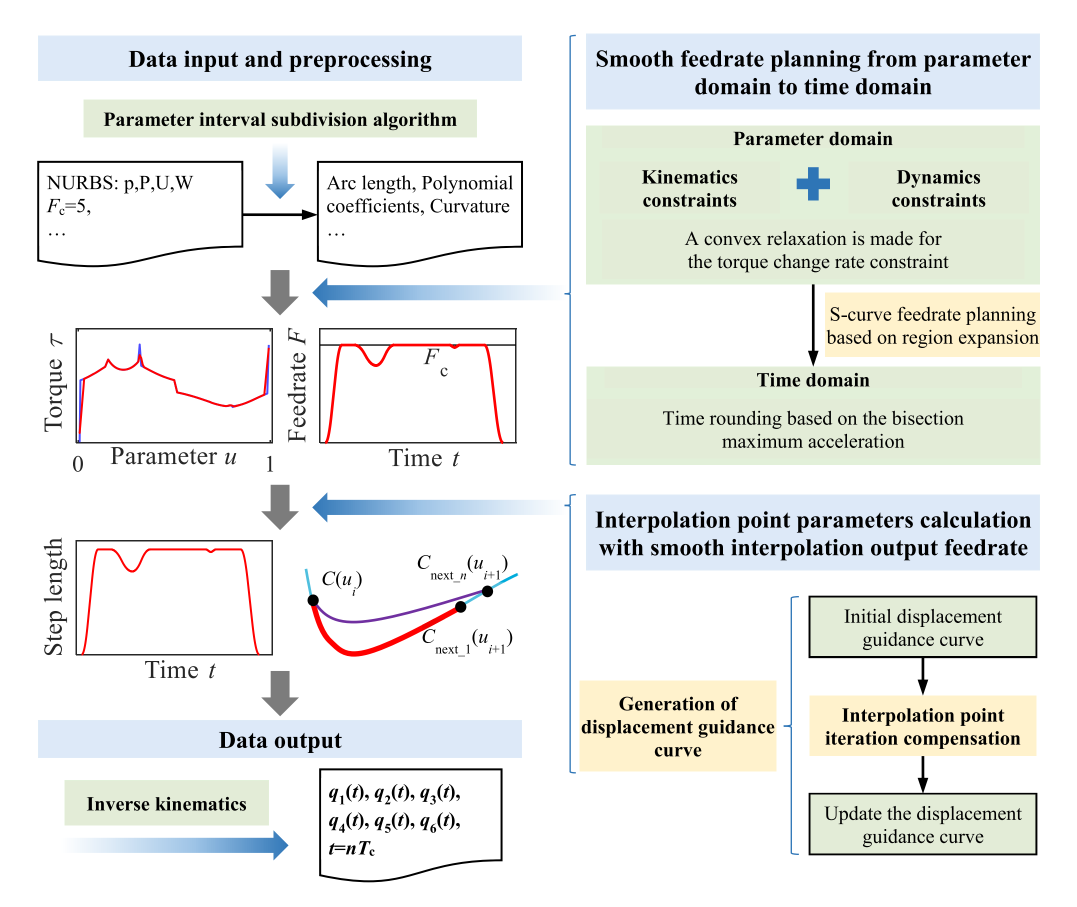


Fig. 1. The schematic diagram of the proposed ISSIOF strategy.

the method that does not consider this constraint. The other graph shows the relationship between time and planned feedrate, with the red curve representing the time-domain feedrate obtained in this stage. This graph illustrates that the above steps can ensure the planned feedrate is smooth after transitioning from the parameter domain to the time domain.

The interpolation point parameter calculation with smooth interpolation output feedrate is as follows: According to the aforementioned feedrate planning results, a relationship curve between the corresponding interpolation period and displacement is generated as a displacement guidance curve. The ideal interpolation step length for this interpolation period is calculated during interpolation. It is determined by subtracting the displacement of the current interpolation period from that of the previous interpolation period on the displacement guidance curve. To ensure that the actual interpolation output step length matches the ideal step length, interpolation point iteration compensation is conducted to generate the actual interpolation points. The guidance curve is updated accordingly, considering the deviation between the actual interpolation output path of the robot and the ideal path. The above steps ensure the interpolation output feedrate is smooth. On the left side of Fig. 1, there are two graphs corresponding to the interpolation point parameter calculation stage. One graph shows the relationship between time and step length, with the red curve being the displacement guidance curve for the proposed method. The other graph is a schematic diagram of interpolation point iteration compensation, where the blue curve represents the ideal path, the red curve represents the ideal path in the current period, and the purple curve represents the actual interpolated path.

In the data output stage, the inverse kinematics is applied based on the interpolation points generated by the aforementioned interpolation. This process results in the output of robot joint space commands.

3. Smooth feedrate planning from parameter domain to time domain

When the direct transcription method for robot feedrate planning is directly converted to the time domain, the planned feedrate is under-smooth and there are round-off errors. To solve this problem, the proposed method takes into account the constraint of the torque change rate during feedrate planning in the parameter domain, and ensures the smoothness of the planned feedrate in the time domain through S-curve feedrate planning and time rounding methods.

3.1. Parameter domain feedrate planning considering dynamics constraints

3.1.1. Kinematics constraints

The end-effector motion path of the robot is a NURBS curve, which can be represented by the parameter u(0–1). During feedrate planning for this path, both the feedrate along the path and the planning time are unknowns. As a result, the first and second derivatives of the joint angle $\boldsymbol{q}$ with respect to time, which are required for the kinematic and subsequent dynamic constraints, are also unknown. To solve this problem, $\boldsymbol{q}$ can be expressed in the form of $\boldsymbol{q}(u)$. Then, the first and second derivatives of $\boldsymbol{q}$ with respect to time can be written in the following parametric forms according to the chain rule:

$$\dot{\boldsymbol{q}}(u) = \boldsymbol{q}'(u)\dot{u} \tag{1}$$

$$\ddot{\boldsymbol{q}}(u) = \boldsymbol{q}'(u)\ddot{u} + \boldsymbol{q}''(u)\dot{u}^2 \tag{2}$$

The robot kinematics constraints considered in this study include the velocity constraints of the robot joint and the end-effector velocity constraints. The velocity constraints of the robot joint are expressed in pseudo-velocity($\dot{u}$) form as follows:

$$\dot{\boldsymbol{q}}_{\min} \leq \boldsymbol{q}'(u)\dot{u} \leq \dot{\boldsymbol{q}}_{\max} \tag{3}$$

where $\dot{\boldsymbol{q}}_{\min}$ and $\dot{\boldsymbol{q}}_{\max}$ denote the velocity bounds of the robot's joint.

Based on the velocity bounds of each joint of the robot, the max end-effector velocity $\boldsymbol{V}_{\mathrm{qj}}(u)$ is calculated as follows:

$$\boldsymbol{v}(u) = \boldsymbol{P}'(u)\dot{u} = \boldsymbol{J}(u)\dot{\boldsymbol{q}}(u) \tag{4}$$

$$\dot{\boldsymbol{q}}_{\mathrm{umax}}(u) = \min\left(\dot{\boldsymbol{q}}_{\max}/\mathrm{abs}\left(\boldsymbol{J}^{-1}(u)\boldsymbol{P}'(u)\right)\right)\boldsymbol{J}^{-1}(u)\boldsymbol{P}'(u) \tag{5}$$

$$\boldsymbol{V}_{\mathrm{qj}}(u) = \boldsymbol{J}(u)\dot{\boldsymbol{q}}_{\mathrm{umax}}(u) \tag{6}$$

where $\boldsymbol{v}(u)$ represents the velocity of the robot's end-effector control point, $\boldsymbol{P}(u)$ signifies the coordinates of the curve, $\boldsymbol{J}(u)$ is the robot's Jacobian matrix about the velocity, and $\min(\cdot)$ is a math operation that finds the smallest element in a vector.

In path tracking tasks, it is important to make sure that the feedrate of the robot's end-effector control point remains as close as possible to the recommended reference feedrate V_{ref}. The smaller value of V_{qjn} (the vector module of the max end-effector velocity) and V_{ref} is then assigned to the feedrate command F_{C}. The following formula is then used to express F_{C} in the form of $\dot{u}$, yielding the pseudo-velocity dup:

$$\boldsymbol{v}(u) = \boldsymbol{P}'(u)\dot{u} \tag{7}$$

$$\dot{u} = \text{Norm}(\boldsymbol{v}(u))/\text{Norm}(\boldsymbol{P}'(u)) \tag{8}$$

$$dup = F_C/\text{Norm}(\boldsymbol{P}'(u)) \tag{9}$$

where Norm(·) is used to perform the vector modulus operation.

3.1.2. Dynamics constraints

The robot dynamic equations without considering friction in the time domain are as follows:

$$\boldsymbol{\tau}(t) = \boldsymbol{M}(\boldsymbol{q}(t))\ddot{\boldsymbol{q}}(t) + \boldsymbol{C}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\dot{\boldsymbol{q}}(t) + \boldsymbol{G}(\boldsymbol{q}(t)) \tag{10}$$

Where $\boldsymbol{\tau}$ is the generalized joint torque, $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}$ represents the centrifugal and Coriolis matrix, $\boldsymbol{G}$ is the gravitational torque matrix.

Substitute formulas (1), (2) into formula (10), and the above robot dynamic equations will be transformed into the parameter domain:

$$\boldsymbol{\tau}(u) = \boldsymbol{M}_u(u)\ddot{u} + \boldsymbol{C}_u(u)\dot{u}^2 + \boldsymbol{G}_u(u) \tag{11}$$

where,

$$\boldsymbol{M}_u(u) = \boldsymbol{M}(\boldsymbol{q}(u))\boldsymbol{q}'(u) \tag{12}$$

$$\boldsymbol{C}_u(u) = \boldsymbol{M}(\boldsymbol{q}(u))\boldsymbol{q}''(u) + \boldsymbol{C}(\boldsymbol{q}(u), \boldsymbol{q}'(u))\boldsymbol{q}'(u) \tag{13}$$

$$\boldsymbol{G}_u(u) = \boldsymbol{G}(\boldsymbol{q}(u)) \tag{14}$$

The robot dynamics constraints considered in this study include the joint torque constraints of the robot and the torque change rate constraints. The joint torque constraints of the robot can be represented in parameter u as follows:

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{M}_u(u)\ddot{u} + \boldsymbol{C}_u(u)\dot{u}^2 + \boldsymbol{G}_u(u) \le \boldsymbol{\tau}_{\max} \tag{15}$$

where $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$ is the torque bound of the robot's joint.

To further ensure the smoothness of the robot's end-effector motion, the torque change rate constraint will be considered being added. The derivative of the Eq. (15) yields the torque change rate constraint:

$$\begin{aligned} &\dot{\boldsymbol{\tau}}_{\min} \le \dot{\boldsymbol{M}}_u\ddot{u} + \boldsymbol{M}_u\dddot{u} + \dot{C}_u\dot{u}^2 + 2\dot{u}\boldsymbol{C}_u\ddot{u} + \dot{\boldsymbol{G}}_u \le \dot{\boldsymbol{\tau}}_{\max} \\ \Leftrightarrow\ &\dot{\boldsymbol{\tau}}_{\min} \le \boldsymbol{M}_u\dddot{u} + \dot{\boldsymbol{M}}_u\ddot{u} + 2\boldsymbol{C}_u\dot{u}\ddot{u} + \dot{\boldsymbol{C}}_u\dot{u}^2 + \dot{\boldsymbol{G}}_u \le \dot{\boldsymbol{\tau}}_{\max} \end{aligned} \tag{16}$$

From formula (16), the results show that the rate of torque change is non-convex, which makes it difficult to solve the smooth minimum time feedrate planning problem. To convert formula (16) into a convex form, it is rewritten as:

$$\dot{\boldsymbol{\tau}}_{\min} \le \dot{\boldsymbol{\tau}} \le \dot{\boldsymbol{\tau}}_{\max} \Leftrightarrow |\dot{\boldsymbol{\tau}}| \le \dot{\boldsymbol{\tau}}_{\max} \Leftrightarrow |\boldsymbol{\tau}'\dot{u}| \le \dot{\boldsymbol{\tau}}_{\max} \tag{17}$$

Since the bound of pseudo-velocity is $0 \le \dot{u} \le \dot{u}_{\max} \Leftrightarrow |\dot{u}| \le \dot{u}_{\max}$. Thus, similar to the approach in [45], the upper limit value of the torque change rate can be expressed as:

$$|\dot{u}| \le \dot{u}_{\max} \Leftrightarrow |\boldsymbol{\tau}'\dot{u}| \le |\boldsymbol{\tau}'|\dot{u}_{\max} \Leftrightarrow |\dot{\boldsymbol{\tau}}| \le |\boldsymbol{\tau}'|\dot{u}_{\max} \tag{18}$$

According to the formula (18), if this upper limit is restricted within the range of $\dot{\boldsymbol{\tau}}_{\max}$, then the torque change rate will also meet the requirements. Thus, a convex relaxation is made for the torque change rate constraint as follows:

$$|\dot{\boldsymbol{\tau}}| \le |\boldsymbol{\tau}'|\dot{u}_{\max} \le \dot{\boldsymbol{\tau}}_{\max} \Leftrightarrow |\boldsymbol{\tau}'| \le \dot{\boldsymbol{\tau}}_{\max}/\dot{u}_{\max} \Leftrightarrow |\boldsymbol{\tau}'| \le \boldsymbol{\tau}'_{\max} \tag{19}$$

where $\boldsymbol{\tau}'_{\max}$ is the relaxation torque change rate constraint.

By imposing constraints on the derivative with respect to the parameter, i.e., the relaxation torque change rate, we naturally achieve an equivalent effect of constraining the torque change rate in the time domain.

Based on the direct transcription approach [20], the parameter domain feedrate planning considering dynamics constraints can be rewritten as:

$$\min : \sum_{k=0}^{k-1} 2\Delta u^k d^k$$

$$\text{s.t.}\begin{cases} \boldsymbol{\tau}_{\min} \le \boldsymbol{M}_u\left(u^{k+1/2}\right)a^k + \boldsymbol{C}_u\left(u^{k+1/2}\right)b^{k+1/2} + \boldsymbol{G}_u\left(u^{k+1/2}\right) \le \boldsymbol{\tau}_{\max} \\ \left| \begin{array}{c} \boldsymbol{M}_u\left(u^{k+\frac{3}{2}}\right)a^{k+1} + \boldsymbol{C}_u\left(u^{k+\frac{3}{2}}\right)b^{k+\frac{3}{2}} + \boldsymbol{G}_u\left(u^{k+\frac{3}{2}}\right) - \\ \boldsymbol{M}_u\left(u^{k+\frac{1}{2}}\right)a^{k} - \boldsymbol{C}_u\left(u^{k+\frac{1}{2}}\right)b^{k+\frac{1}{2}} - \boldsymbol{G}_u\left(u^{k+\frac{1}{2}}\right) \end{array} \right| / \Delta u^k \le \boldsymbol{\tau}'_{\max} \\ 0 \le b^k \le dup_k{}^2 \\ b^{k+1} - b^k = 2a^k \Delta u^k \\ b^0 = \dot{u}_0{}^2, b^k = \dot{u}_T{}^2 \\ \left\| \begin{array}{c} 2 \\ c^{k+1} + c^k - d^k \end{array} \right\|_2 \le c^{k+1} + c^k + d^k \\ \left\| \begin{array}{c} 2c^k \\ b^k - 1 \end{array} \right\|_2 \le b^k + 1 \end{cases} \tag{20}$$

where,

$$a(u) = \ddot{u} \tag{21}$$

$$b(u) = \dot{u}^2 \tag{22}$$

$$\frac{1}{\sqrt{b^{k+1}} + \sqrt{b^k}} \le d^k \tag{23}$$

$$c^k \le b^k \tag{24}$$

The above formula (20) belongs to the second-order cone programming problem, and many solvers can solve this problem, such as GUROBI, ECOS, SDPT3, SEDUMI, etc. Due to the strong performance, and high stability of the GUROBI solver, the academic version of GUROBI is selected as the solver for this problem.

The flow chart of the parameter domain feedrate planning considering dynamics constraints is depicted in Fig. 2. The blue curve in the first figure on the right side of the flow chart represents the recommended feedrate V_{ref}, and the magenta curve represents the max end-effector velocity V_{qjn}, with its minimum value being V_{mqjn}. The red curve in the second figure on the right represents the joint torque considering dynamics constraints, while the blue curve represents the joint torque without considering dynamics constraints. The red curve in the third figure on the right indicates the feedrate command F_{C}, and the blue curve represents the planned feedrate considering both kinematics and dynamics constraints. The planned feedrate curve is divided into regions based on the feedrate states: acceleration, constant feedrate, or deceleration, as shown in the last figure on the right. In the figure, the red curve denotes the acceleration region, the black curve denotes the constant feedrate region, and the blue curve denotes the deceleration region.

3.2. Time domain S-curve feedrate planning based on region expansion

While obtaining the planned feedrate in Section 3.1, each period Δt between neighbouring parameters of the NURBS curve can be obtained as follows:

$$\Delta t^k = 2\Delta u^k / \left(\dot{u}^{k+1} + \dot{u}^k\right) \tag{25}$$

where k is the kth member of the NURBS parameters.

Once the time variable t is computed, the feedrate $v(u)$ planned in the parameter domain can be transformed to the time domain $v(t)$. Based on the two-point central difference as well as the three-point central difference, the acceleration a_t and jerk j_t in the time domain are calculated as follows:

$$a_t^k = \left(v^{k+1} - v^{k-1}\right) / \left(2\Delta t^k\right) \tag{26}$$

$$j_t^k = \left(v^{k-1} - 2v^k + v^{k+1}\right) / \left(\Delta t^k\right)^2 \tag{27}$$

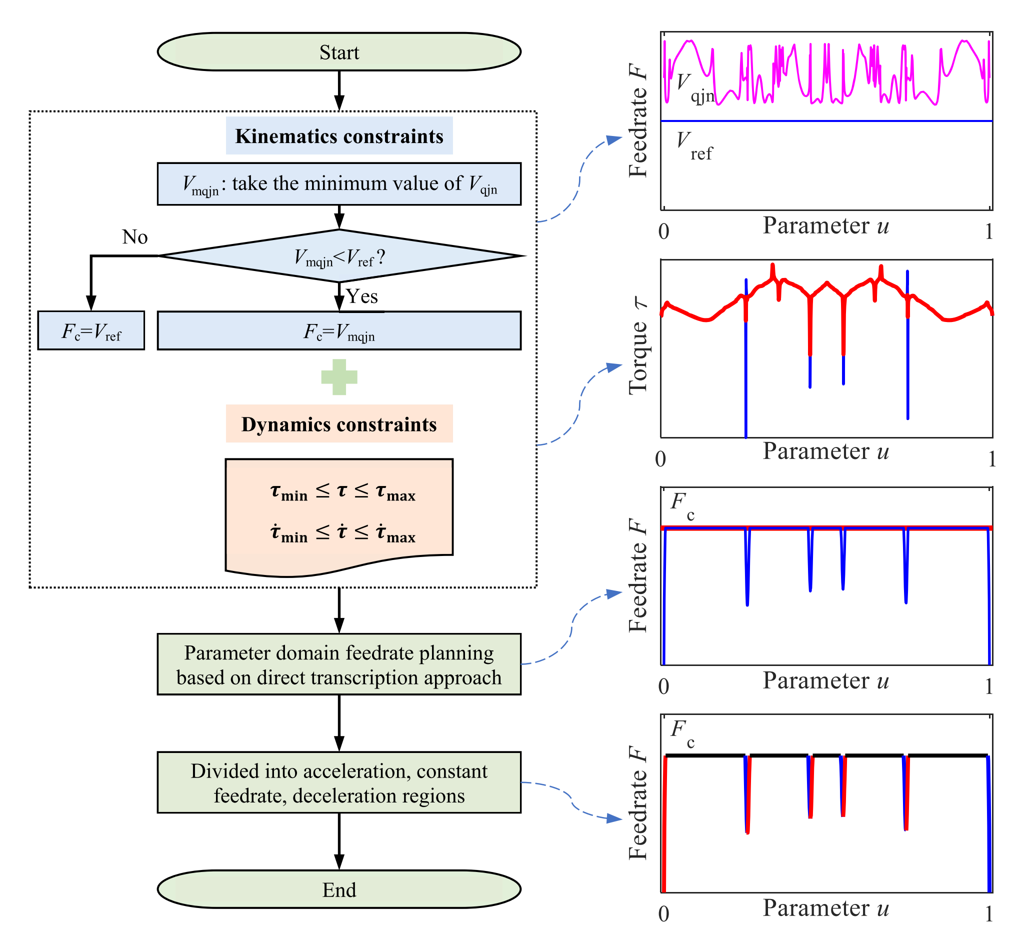


Fig. 2. The flow chart of the parameter domain feedrate planning considering dynamics constraints.

The flow chart of the time domain S-curve feedrate planning based on region expansion is shown in Fig. 3. In the acceleration and deceleration regions, the minimum acceleration and jerk within the region are selected as the conservative maximum acceleration and maximum jerk. If the current region is an acceleration region, the maximum ending feedrate is calculated for acceleration to the last point of the acceleration region. If the current region is a deceleration region, the maximum starting feedrate is calculated for reverse acceleration to the starting point of the deceleration region. Judge whether the maximum ending feedrate is greater than the region ending feedrate or whether the maximum starting feedrate is greater than the region starting feedrate. If not satisfied in the acceleration region, the region is expanded forwards until there is enough distance to accelerate to the region ending feedrate. If not satisfied in the deceleration region, the region is extended backwards until there is enough distance to accelerate in reverse to the region starting feedrate. If satisfied, S-curve feedrate planning is performed according to the region starting feedrate, region ending feedrate, conservative maximum acceleration, conservative maximum jerk, and current region length. Since expanding the acceleration region forwards and expanding the deceleration region backwards both reduce the constant feedrate region. It is important to update the starting and ending points of the constant feedrate region. Finally, the planned feedrate of each region is connected to obtain a smooth time domain planned feedrate.

3.3. Time rounding based on the bisection maximum acceleration

After the S-curve feedrate planning in Section 3.2, the time for the acceleration and deceleration region is increased, and the time for the constant feedrate region is also changed. Although the above method ensures that the planned feedrate is smooth in the time domain, there is still a round-off error problem for the total planned time. This will also affect the smoothness of the motion. To solve this problem, time rounding based on the bisection maximum acceleration is proposed, as detailed below. The adjacent acceleration, constant feedrate, and deceleration regions are combined into one interval. The total time of each interval T_{qj} is calculated as follows:

$$T_{qj}^{i} = T_{a}^{i} + T_{c}^{i} + T_{d}^{i} \tag{28}$$

where i denotes the ith interval, T_a, T_c, and T_d are the acceleration region time, constant feedrate region time, and deceleration region time, respectively.

The total time of each interval is calculated to determine whether it can be strictly divided by the interpolation period. If it cannot be strictly divided, it is rounded up as follows:

$$T_{\text{qjz}}^{i} = T_{\text{qj}}^{i} + \Delta t^{i} \tag{29}$$

where T_{qjz} and Δt are the rounded total time of each interval and the additional time required for rounding, respectively.

The additional time required for rounding is evenly distributed to the acceleration and deceleration regions of this interval:

$$T_{\text{az}}^{i} = T_{\text{a}}^{i} + \Delta t^{i}/2 \tag{30}$$

$$T_{\text{dz}}^{i} = T_{\text{d}}^{i} + \Delta t^{i}/2 \tag{31}$$

Within the acceleration and deceleration regions, the initial maximum acceleration is used as the upper limit of the maximum acceleration a_{maxup}. The lower limit of the maximum acceleration a_{maxlow} is calculated as follows:

$$a_{\text{maxlow}}^{i} = a_{\text{maxup}}^{i} \tag{32}$$

$$a_{\text{maxlow}}^{i} = a_{\text{maxlow}}^{i} \cdot s \tag{33}$$

where s ($0 < s < 1$) denotes the scaling factor.

The S-curve feedrate planning is performed based on the a_{maxlow} obtained from Eq. (33), and the time T_{low} in the region is calculated. If $T_{\text{low}}>T_{\text{az}}$(or T_{dz}), then a_{maxlow} is calculated. Otherwise, Eq. (33) is repeated to calculate a_{maxlow}, and S-curve feedrate planning is performed until $T_{\text{low}}>T_{\text{az}}$(or T_{dz}). Through the bisection method, an appropriate maximum acceleration a_{maxzj} is found so

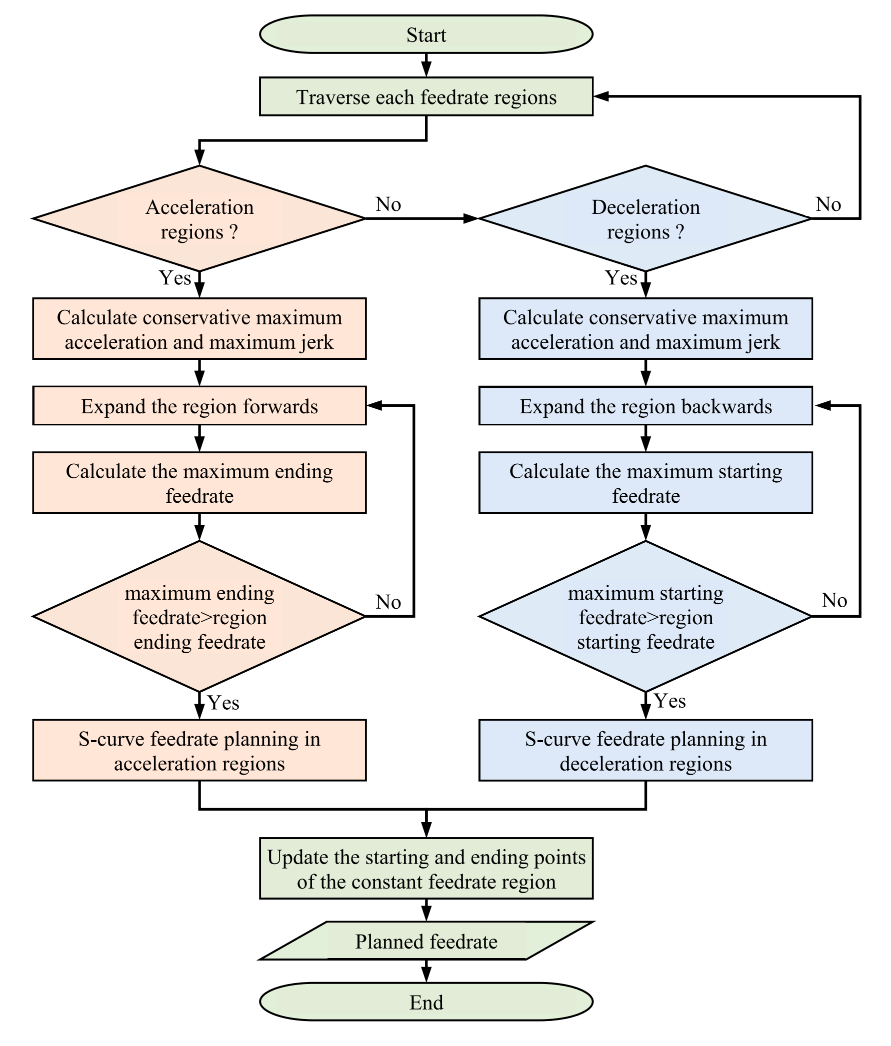


Fig. 3. The flow chart of the time domain S-curve feedrate planning based on region expansion.

that the time of S-curve feedrate planning T_s is equal to the required time of the acceleration or deceleration region T_{az}(or T_{dz}). The flow chart of the time rounding based on the bisection maximum acceleration is shown in Fig. 4.

4. Interpolation point parameters calculation with smooth interpolation output feedrate

The deviation between the actual interpolation path and the ideal path is illustrated in Fig. 5. The upper left part of Fig. 5 shows the coarse interpolation stage, with the black curve representing the ideal path and P_1 and P_2 representing the current and next coarse interpolation points, respectively. At this stage, the P_2 point is obtained based on the coarse interpolation algorithm, and its corresponding joint angle is obtained through inverse kinematics. The lower part of Fig. 5 is the fine interpolation stage, where the black solid dots represent the joint angles corresponding to P_1 and P_2, and the black curve represents the joint ideal path corresponding to the robot workspace ideal path. The red dashed line represents the joint command curve obtained through fine interpolation, and the black circle on it represents the joint command corresponding to the fine interpolation cycle(T_j). At this stage, joint interpolation commands are obtained based on the PVT fine interpolation algorithm. The upper right part of Fig. 5 shows the robot actual interpolation path during a coarse interpolation cycle(T_c). The red dashed line represents the actual interpolation path for the end-effector, and the black circle represents the end-effector fine interpolation point corresponding to the fine interpolation cycle. Fig. 5 illustrates that the deviation between the actual interpolation path and the ideal path is primarily due to the non-linear motion relationship between the robot's joint and the end-effector, as well as the imperfect tracking of ideal joint angles during the fine interpolation process.

To address the issue of unsmooth interpolation output feedrate caused by the actual interpolation path deviating from the ideal

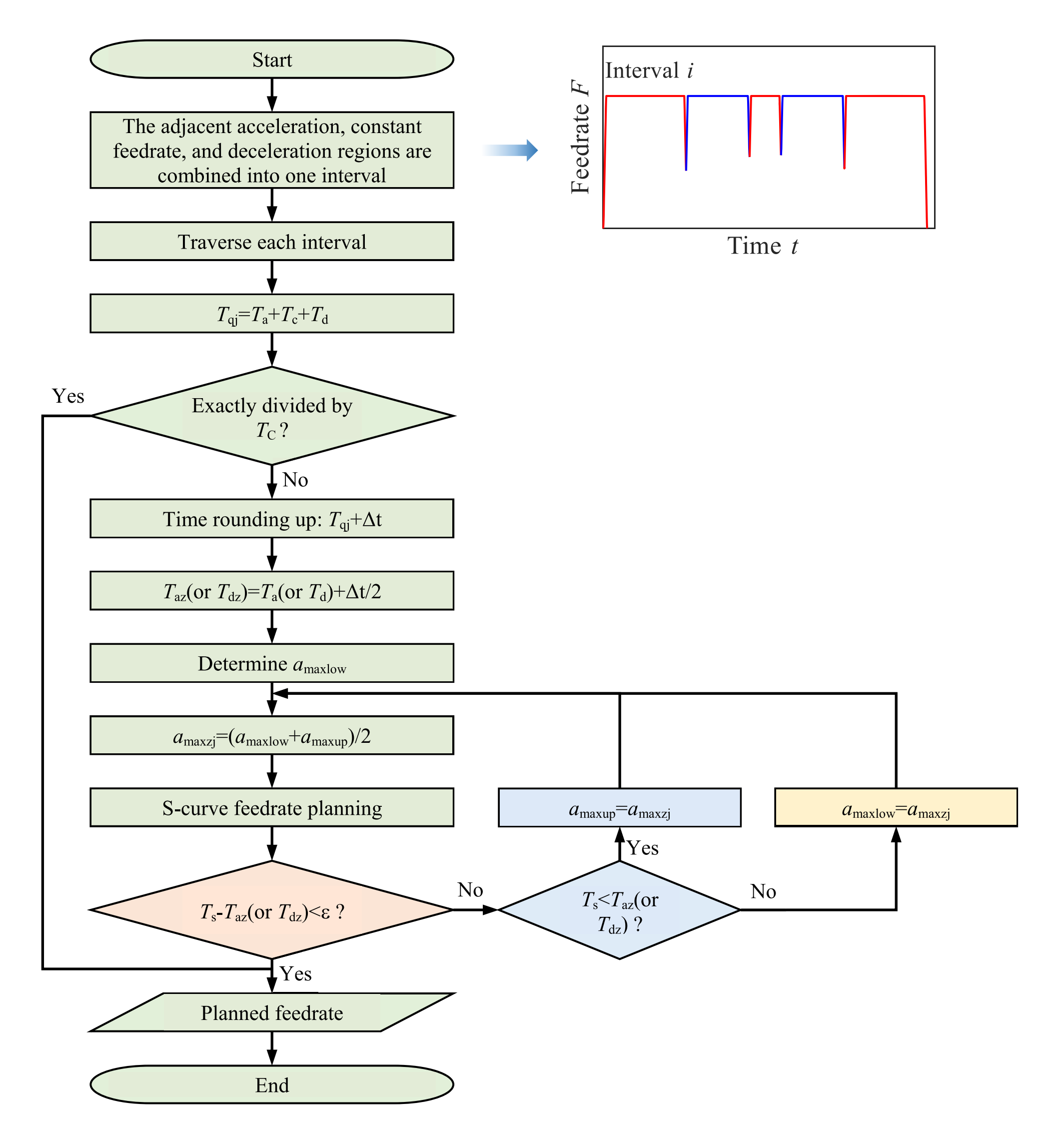


Fig. 4. The flow chart of the time rounding based on the bisection maximum acceleration.

path, a displacement guidance curve is generated during the interpolation stage. This guidance curve is combined with interpolation point iteration compensation to ensure that the actual interpolation output step length matches the guidance curve.

4.1. Generation of displacement guidance curve

The smooth feedrate planning method in Section 3 can generate the relationship curve $d(t)$ between displacement and interpolation time while generating the planned feedrate. This curve is served as the initial displacement guidance curve for guiding the subsequent interpolation process. During interpolation, the ideal interpolation step length l_{step} for this interpolation period is calculated. l_{step} is determined by subtracting the displacement of the current interpolation period from that of the previous interpolation period on the displacement guidance curve.

$$l_{\mathrm{step}}^{j} = d\left(t^{j+1}\right) - d\left(t^{j}\right) \tag{34}$$

where j denotes the jth interpolation period.

In order to further smooth the output feedrate of robot NURBS curve interpolation, it is necessary to ensure that the actual interpolation output step length matches the ideal step length. To achieve this goal, interpolation point iteration compensation is conducted to generate the desired actual interpolation points. The specific steps are described in Section 4.2. Because of the deviation of the robot's actual interpolation path from the ideal path, there may be two situations after interpolation point iteration compensation. One situation is that before the guidance curve is completed, the current interpolation point exceeds the last point of the curve. Consequently, the remaining guidance curve loses its guiding role. Another situation is that after the guidance curve is completed, the current interpolation point has not yet reached the last point of the curve. Thus, the guidance curve needs to be extended.

The actual interpolation point d_{c}^{j} is expressed as the total displacement from the first point of the curve to the current interpolation point. The last point of this curve is represented by the total displacement of the curve d_{t}. Within the guidance curve period, it is judged in each period whether the actual interpolation point exceeds the last point of the curve by the following equation:

$$\rho = d_{\mathrm{c}}^{j} - d_{\mathrm{t}} \tag{35}$$

where ρ represents the judgment term, and the value of ρ less than zero means that the current actual interpolation point does not exceed the last point.

If the current actual interpolation point exceeds the last point, the step length of this period is updated just to reach the last point. The program ends prematurely following this update, and the remaining guidance curve loses its guiding role. The step length of this period is:

$$l_{\mathrm{step}}^{j} = d_{\mathrm{t}} - d_{\mathrm{c}}^{j} \tag{36}$$

If the actual interpolation point does not exceed the last point of the curve, and the maximum period member m of the guidance curve is reached. When this happens, it is necessary to extend the period of the guidance curve. This ensures the step length of subsequent periods equals to that of the last period of the initial displacement guidance curve. The step length of subsequent periods is:

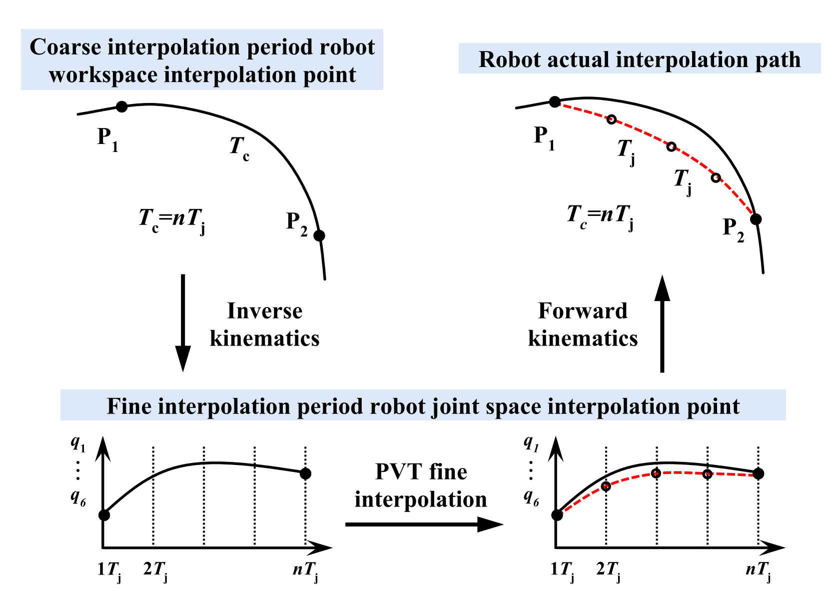


Fig. 5. The schematic diagram of the robot actual interpolation path.

$$l_{\text{step}}^{m+n} = d_{\text{c}}^{m} - d_{\text{c}}^{m-1} \tag{37}$$

where n represents the number of subsequent periods.

In extended periods, an interpolation point generated in a certain period may exceed the last point of the curve. If this happens, the step length of this period should also be updated just to reach the last point, and then the program ends. The step length of this period is:

$$l_{\text{step}}^{end} = d_{\text{t}} - d_{\text{c}}^{m+n} \tag{38}$$

The flow chart of the generation of displacement guidance curve is shown in Fig. 6.

4.2. Interpolation point iteration compensation

During interpolation, the ideal step length l_{step} is calculated through the displacement guidance curve. On this basis, the next interpolation point position $C_{\text{next_1}}(u_{i+1})$ is calculated by the fitting interpolation method, which is based on cubic polynomial [3]:

$$l_{\text{d}} = l_{\text{p}} + l_{\text{step}}^{j} \tag{39}$$

$$C_{\text{next_1}}(u_{i+1}) = c_1(l_{\text{d}})^3 + c_2(l_{\text{d}})^2 + c_3 l_{\text{d}} + c_4 \tag{40}$$

where l_{d} represents the desired arc length on the curve segment obtained in the preprocessing stage, l_{p} represents the desired arc length of the previous interpolation period, and c_1-c_4 represent the polynomial coefficients obtained in the preprocessing stage.

Because the actual interpolation path deviates from the ideal path during the current period, the current actual interpolation step length is not equal to the ideal step length. The difference ΔS between the current actual interpolation step length l_{actual} and the ideal interpolation step length l_{step} is:

$$\Delta S = l_{\text{actual}}^{j} - l_{\text{step}}^{j} \tag{41}$$

By subtracting ΔS from the ideal path, the next point $C_{\text{next_2}}(u_{i+1})$ on the ideal path is obtained. The corresponding actual interpolation step length at this time is calculated subsequently:

$$l_{\text{d}} = l_{\text{d}} - \Delta S \tag{42}$$

$$C_{\text{next_2}}(u_{i+1}) = c_1(l_{\text{d}})^3 + c_2(l_{\text{d}})^2 + c_3 l_{\text{d}} + c_4 \tag{43}$$

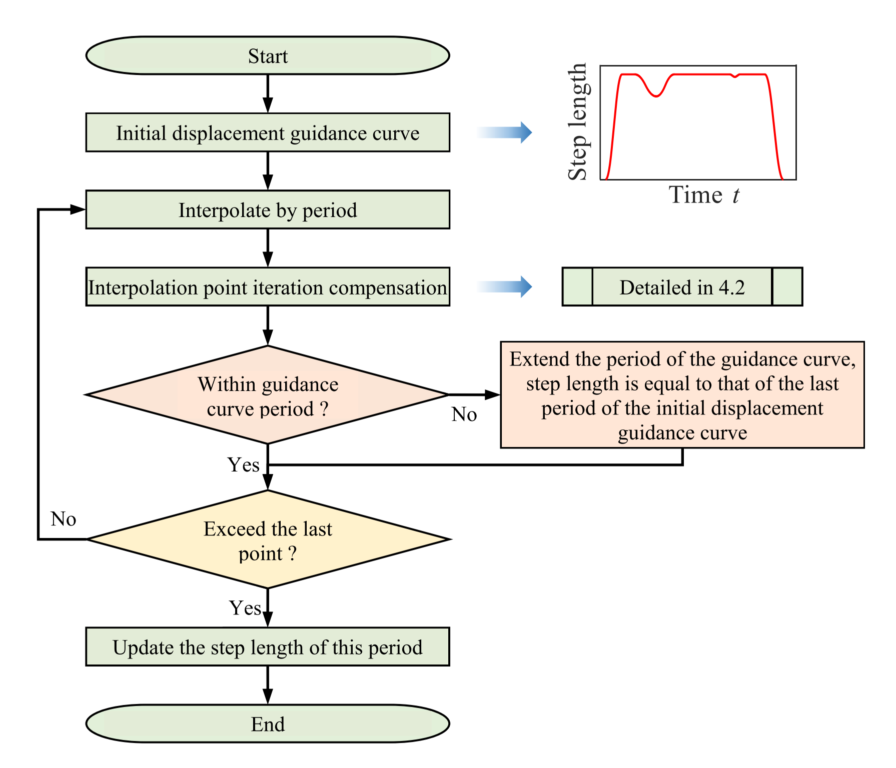


Fig. 6. The flow chart of the generation of displacement guidance curve.

The above steps are iteratively executed until the actual interpolation step length matches the ideal interpolation step length, thereby determining the interpolation point $C_{next_n}(u_{i+1})$ that meets the requirements. The schematic diagram of the interpolation point iteration compensation is shown in Fig. 7. In the figure, the blue curve represents the ideal path. The red curve represents the ideal path in the current period. The black curve represents the actual interpolation path. And the purple curve represents the actual interpolation path obtained from the nth iteration.

In order to accelerate the convergence speed of the above iterative compensation, the arc length of the actual interpolation point in the current segment is recorded during the iteration process. When the ΔS is positive, the current arc length is recorded as the upper limit. The upper limit is updated when a recorded arc length is less than the previously recorded upper limit. Conversely, when the ΔS is negative, the current arc length is recorded as the lower limit. The lower limit is updated if the recorded arc length exceeds the previously recorded lower limit. As the number of iterations increases, the upper and lower limits continue to shrink, increasing the convergence speed. The above interpolation point iterative compensation procedure is illustrated in Fig. 8.

5. Simulation and experimental results

In this section, simulations and experiments are conducted on a robot with 6 degrees of freedom (as shown in Fig. 9). The robot's D-H parameters are provided in Table 1. A WM-shaped NURBS curve is employed for the simulations and flank milling experiments. To demonstrate the efficacy of the proposed ISSIOF, the parameter domain feedrate planning considering dynamics constraints in Section 3.1 and the fitting interpolation method based on cubic polynomial [3] are combined. This yields the NURBS curve interpolation strategy with under-smoothed interpolation output feedrate (ISUSIOF), which is subsequently compared to the proposed ISSIOF.

5.1. Simulation

Fig. 10(a) depicts the WM-shaped curve employed in the simulation and experiment. Some areas on the WM-shaped curve are marked with letters A to F for subsequent simulation and experimental analysis. The NURBS parameters are given as follows: degree $p = 2$, control points (mm) $P = [(0, 0), (12, -14), (24, -4), (36, -14), (48, 6), (60, -2), (72, 6), (84, -16)]$, knot vector $U = [0, 0, 0, 0.15, 0.3, 0.5, 0.7, 0.8, 1, 1, 1]$, and weight vector $W = [1, 1.5, 1.5, 1.5, 2, 2, 2, 1]$. Moreover, (b) illustrates the placement of this curve within the robot's base coordinate system, and the three-dimensional coordinate of the start point is (1.165748 m, −0.012 m, 0.4392 m). It should be noted that the coordinates of the start point in the simulation are consistent with those in the subsequent experiments. The relevant parameters employed in the study are as follows: the coarse interpolation period T_c is 4 ms, and the fine interpolation period T_j is 1 ms. The joint velocity constraints for each joint are [0.1069, 0.1415, 0.1298, 0.4083, 0.3142, 0.3927] rad/s, and the torque constraints are [135, 102, 52, 8, 10, 4] N·m.

In the case of feedrate commands set at 5 mm/s and 10 mm/s, the proposed ISSIOF and the ISUSIOF are employed for interpolation, with the corresponding interpolation output feedrate being calculated. The simulation results are shown in Figs. 11 and 12. In each figure, (a) represents the interpolation output feedrate along with its local magnification graph, (b) represents the interpolation output acceleration, and (c) represents the interpolation output jerk. The red curve in the figure represents the ISSIOF method, and the blue curve represents the ISUSIOF method. Fig. 13 shows the joint torques before and after considering the torque change rate constraint when the feedrate command is 10 mm/s. Similarly, the simulation results at 5 mm/s exhibit the same trend, differing only in magnitude, and are therefore not presented separately. The letters A-F marked above Figs. 11-13 correspond to the letters marked in Fig. 10. The processing time, path velocity fluctuation, and acceleration and jerk fluctuations corresponding to the two methods, as determined by simulation, are presented in Tables 2, 3, 4, and 5, respectively. The path velocity fluctuation is calculated in the constant feedrate region of each interpolation output feedrate curve. The acceleration and jerk fluctuations are calculated in the constant acceleration regions near points B and F.

As can be seen from the local magnification graph in Figs. 11(a) and 12(a), both the blue and red curves exhibit fluctuations. However, the fluctuation of the blue curve is more obvious. This indicates that the interpolation output feedrate generated by the ISUSIOF method is not smooth, whereas the proposed ISSIOF method can effectively smooth the interpolation output feedrate.

In Fig. 11(b) and 12(b), it can be observed that the maximum values of the blue curve are larger than that of the red curve in each acceleration and deceleration region. The blue curve exhibits slight fluctuations, whereas the red curve is more regular and smoother compared to the blue curve. This indicates that the interpolation output acceleration generated by the ISSIOF method is relatively smaller and smoother.

Fig. 11(c) and 12(c) show that most of the maxima of the blue curve are larger than that of the red curve, and the blue curve exhibits

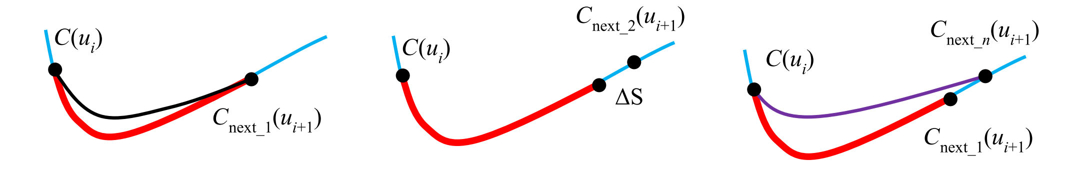


Fig. 7. The diagram of the interpolation point iteration compensation.

Procedure Interpolation point iteration compensation

Input: l_p, l_{step}, tolerance
Output: u_{i+1}
l_{d1} = [], l_{d2} = [] // Initialize lower and upper limit
$l_{actual} = l_{step}$ // Initialize actual step length to desired step length
$l_d = l_p + l_{step}$ // Calculate the desired arc length on the curve segment
while $l_{step} > 0$ **do**
  $\Delta S = l_{actual} - l_{step}$ // Calculate step difference
  // Update lower and upper limits
  if $\Delta S > 0$ **then**
    if l_{d2} is empty or $l_d < l_{d2}$ **then**
      $l_{d2} = l_d$
    end if
  else if $\Delta S < 0$
    if l_{d1} is empty or $l_d > l_{d1}$ **then**
      $l_{d1} = l_d$
    end if
  end if
  // Ensure actual step length is within lower and upper limits
  while $l_d - \Delta S > l_{d2}$ **do**
      $\Delta S = \Delta S / 2$
  end while
  while $l_d - \Delta S < l_{d1}$ do
      $\Delta S = \Delta S / 2$
  end while
  Update desired arc length on the segment according to Eq. (42)
  Update parameter u_{i+1} according to Eq. (43)
  Calculate and update l_{actual}
  // Break loop if step difference is less than tolerance
  if $|l_{actual} - l_{step}|/l_{step}$ < tolerance **then**
    break
  end if
  // Break loop if the last point is reached
  if $u_{i+1} = 1$ **then**
    break
  end if
end while

Fig. 8. The procedure of the interpolation point iteration compensation.

large fluctuations. This shows that the interpolation output jerk generated by the ISSIOF method is smaller and smoother.

As illustrated in Fig. 13, the blue curve exhibits significant sudden changes in the acceleration and deceleration regions. While the red curve also has sudden changes, these are substantially reduced compared to the blue curve. This indicates that the torque required by the robot joints is not smooth when the torque change rate constraint is not considered. When the torque change rate constraint is considered, the required joint torque becomes smoother, with a noticeable effect on the acceleration and deceleration regions.

The aforementioned observations show that the proposed ISSIOF method can effectively reduce the sudden changes of torque for joints. It reduces the maximum values of interpolation output acceleration and jerk. Additionally, it reduces the fluctuations in interpolation output feedrate, acceleration, and jerk, thereby enhancing the smoothness of the robot's end-effector motion.

As can be seen from Table 2, the ISSIOF method does not introduce a significant increase in processing time compared to the ISUSIOF method, indicating that the proposed strategy ensures high efficiency. Tables 3-5 show that the ISSIOF method effectively reduces feedrate, acceleration, and jerk fluctuations compared to the ISUSIOF method, demonstrating the strategy's ability to significantly smooth the interpolation output feedrate.

5.2. Experiment

Flank milling experiments are conducted on a constructed robot platform. The main goal is to test if the ISSIOF method can reduce roughness and contour errors by smoothing the interpolation output feedrate. The robot experimental platform construction is illustrated in Fig. 14. The control cabinet is mainly composed of the drive system and control system. High-performance GSHD servo drives are employed by the robot's drive system. The hardware component of the robot's numerical control system is based on the

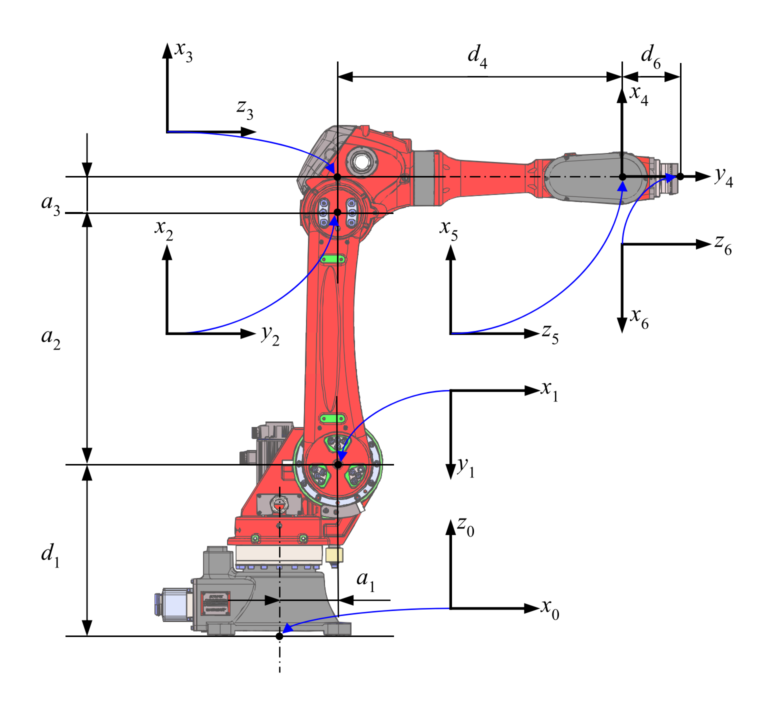


Fig. 9. The diagram of the robot.

Table 1
The D-H parameters of the robot.

Joint	θ(rad)	d(m)	a(m)	α(rad)
1	θ_1	0.4946	0.17	-π/2
2	θ_2(-π/2)	0	0.73	0
3	θ_3	0	0.1	-π/2
4	θ_4	0.8255	0	π/2
5	θ_5	0	0	-π/2
6	θ_6(π)	0.164	0	0

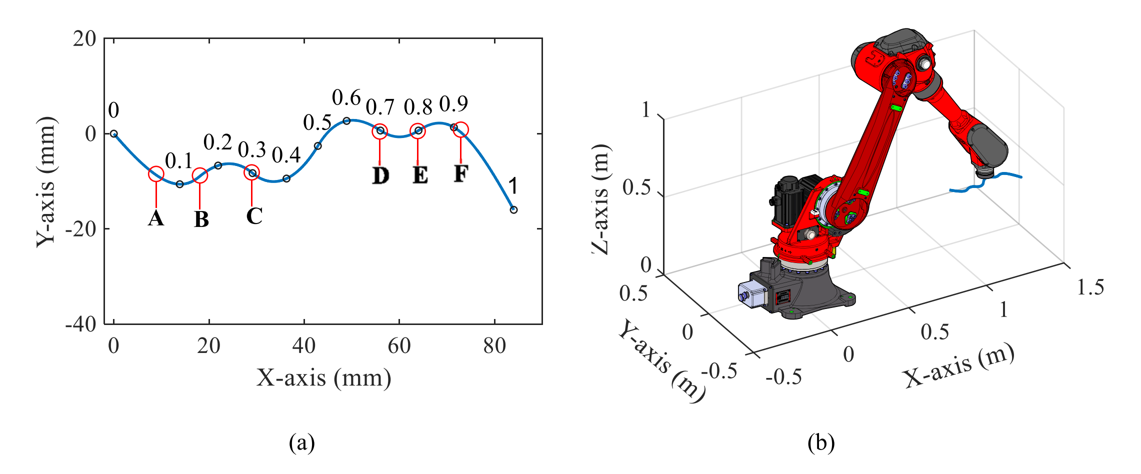


Fig. 10. The schematic of (a) the WM-shaped curve and (b) its placement within the robot's base coordinate system.

OP4510-V2 controller. The software component is realized through Rt-lab and Simulink development environment. The software component of the robot's numerical control system includes both manual and automatic modes. In manual mode, joint space control and workspace control are used for direct operation of the robot joints or end-effector. In automatic mode, the system reads offline joint

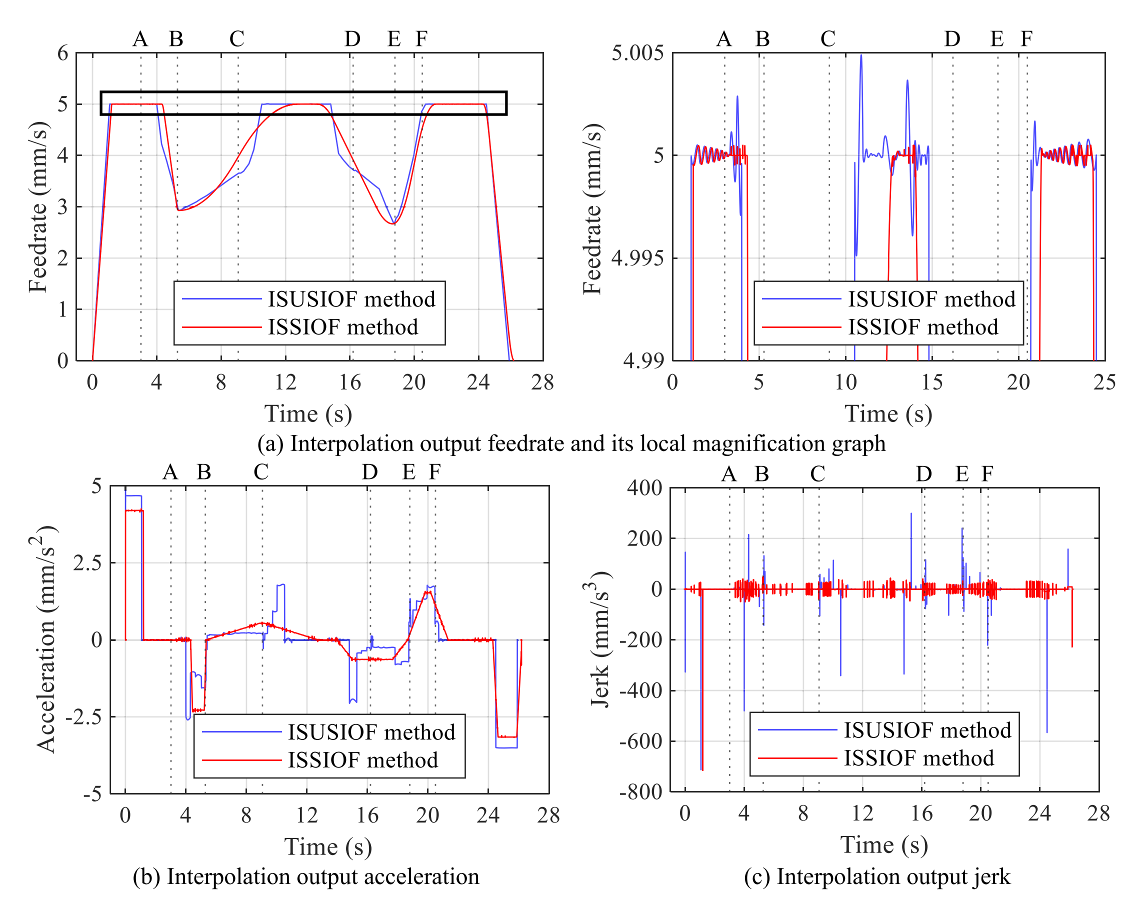


Fig. 11. Simulation results when $F = 5$ mm/s.
(a) Interpolation output feedrate and its local magnification graph, (b) Interpolation output acceleration (c) Interpolation output jerk.

trajectory commands, performs coordinate transformations, and executes the trajectory at any position. The controller generates pulse and direction signals to drive the motors, thus realizing the integration of the proposed method into the robot controller.

A cubic workpiece with dimensions of 200 mm × 86 mm × 35 mm is fixed in a vise. The workpiece material is nylon material MC901. The experimental tool path is the same as that in Fig. 10 of the simulation. A double-flute flat-end mill with a diameter of 6 mm is employed for the flank milling experiment. The electric spindle is installed on the robot as the end-effector, with a power of 1.2 kW. The spindle speed is set to 10,000 r/min, the cutting depth is 2 mm, and the cutting width is 0.5 mm.

To evaluate the performance of the proposed ISSIOF method, flank milling experiments are conducted. Both the ISSIOF and ISUSIOF methods are employed in the case of feedrate commands set as 5 mm/s and 10 mm/s. The experimental results for the feedrate, acceleration, and jerk of the robot's end-effector are shown in Figs. 15 and 16. The feedrate, acceleration, and jerk of the end-effector are obtained by taking the time derivatives of the end-effector position and applying an appropriate smoothing process. The position of the end-effector is calculated through forward kinematics based on joint positions, which are obtained from encoders mounted on the motors. The experimental results for the torque and torque change rate for each robot joint at a feedrate command of 10 mm/s are shown in Fig. 17. The results for 5 mm/s exhibit the same trend, differing only in magnitude, and are therefore not presented separately In Figs. 15-17, the red curve represents the proposed ISSIOF method, while the blue curve represents the ISUSIOF method. The letters A-F marked above Figs. 15-17 correspond to the letters marked in Fig. 10.

As can be seen from Figs. 15(a) and (b) and Figs. 16(a) and (b), the experimental results are consistent with the simulation results. This indicates that the ISSIOF method can effectively smooth the interpolation output feedrate, reduce the maximum value of the interpolation output acceleration, and make the interpolation output acceleration smoother. In Fig. 15(c) and 16(c), the jerk is obtained by taking the third derivative of the robot end-effector's position. This introduces measurement noise that is difficult to eliminate, causing significant fluctuations in both methods. However, the ISSIOF method still reduces the maxima of the interpolation output jerk. As shown in Fig. 17(a) and (b), although both the blue and red curves exhibit fluctuations, the fluctuations in the red curve are noticeably smaller.

The experimental results above demonstrate that the proposed ISSIOF method can effectively smooth the joint torque, reduce the maximum values of the interpolation output acceleration and jerk, and minimize the fluctuations in interpolation output feedrate, acceleration, and jerk. Consequently, it improves the smoothness of the robot end-effector's motion.

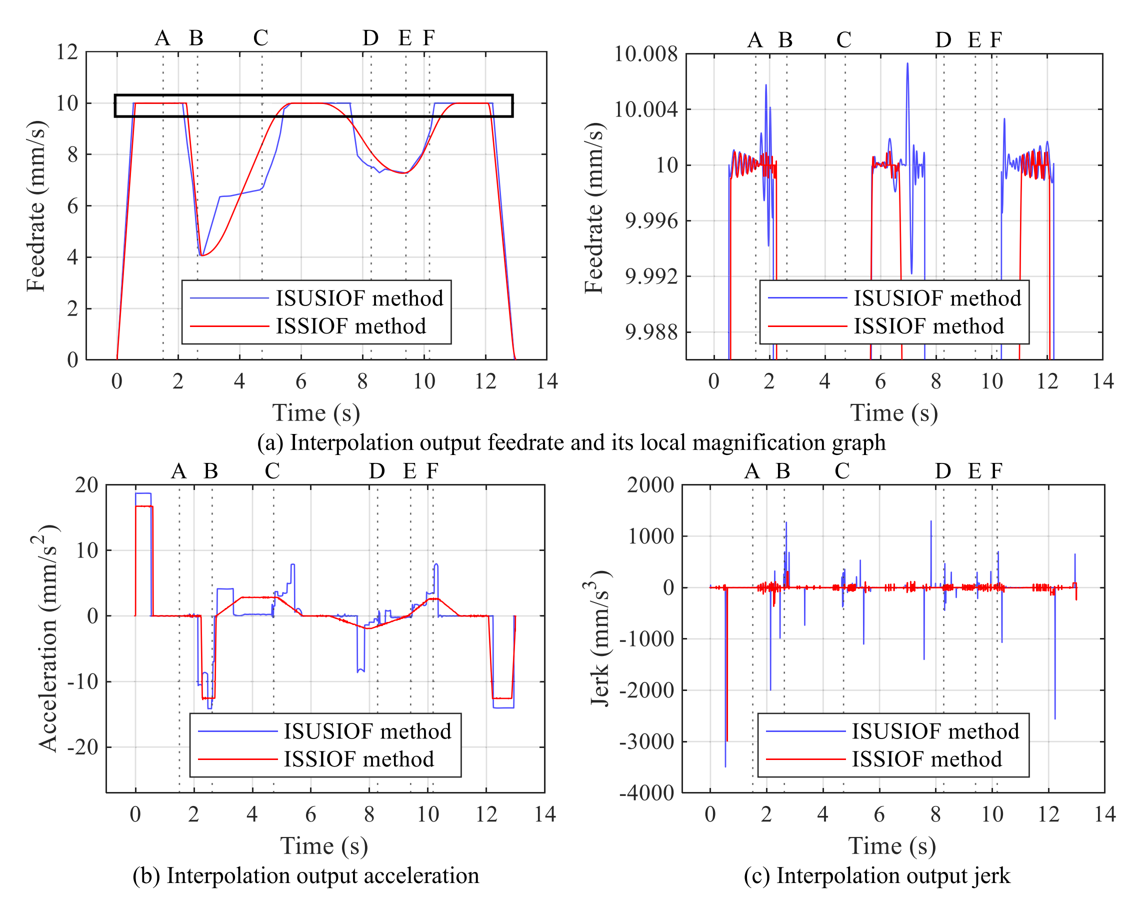


Fig. 12. Simulation results when $F = 10$ mm/s.
(a) Interpolation output feedrate and its local magnification graph, (b) Interpolation output acceleration, (c) Interpolation output jerk.

The results of the flank milling experiment are shown in Fig. 18. The local magnification graph is taken by the Dino-Lite Digital Microscope. One workpiece is used in the experiment. First, the workpiece is rough machined on a machine tool. Then, the ISUSIOF method is employed for flank milling on the robot experimental platform. Subsequently, the workpiece is flipped 180°, and the proposed ISSIOF method is employed for flank milling. In order to quantitatively compare the results of flank milling, the total profile after the flank milling is measured by the Form Talysurf PGI NOVUS measurement system (Fig. 19). The roughness profile obtained by post-processing the measurement data is shown in Fig. 20. The roughness results Ra calculated based on the roughness profile are shown in Table 6. The primary profile obtained by post-processing the measurement data is shown in Fig. 21. Since the actual primary profile is very close to the ideal profile, the contour error in Fig. 21 has been magnified 8 times for easier observation. The total profile is measured by the measurement system over the machined surface. The primary profile and the roughness profile can be obtained by filtering the total profile. According to the ISO 4287 standard, an S-filter is applied to the total profile to suppress high-frequency noise, resulting in the primary profile. Then, a C-filter is applied to the primary profile to suppress longwave components, thus obtaining the roughness profile. Based on the roughness profile, the arithmetic mean deviation is calculated to obtain the value of Ra. The contour error obtained from the total profile is shown in Fig. 22. To calculate the contour error, the measured profile is first aligned with the ideal profile using an optimization method. The optimization variables are two degrees of freedom for translation in the plane and one degree of freedom for rotation. The objective is to minimize the sum of the squared distances between the measured profile and the ideal profile. Once aligned, the shortest distance from each point on the measured profile to the ideal profile is calculated according to the definition of contour error, and this distance represents the contour error. The average contour error and maximum contour error results for both methods are shown in Tables 7 and 8. In Figs. 20-22, the red curve represents the proposed ISSIOF method, while the blue curve represents the ISUSIOF method. The black curve in Fig. 21 represents the ideal profile, which is obtained by offsetting the path by a distance of 3 mm in Fig. 10. The letters A-F marked above Figs. 20-22 correspond to the letters marked in Fig. 10.

As depicted in Fig. 18(a) and (b), the ISUSIOF method exhibits tool marks and irregular textures. In contrast, the proposed ISSIOF method produces significantly smoother and more uniform surfaces. This improvement is attributed to the smoother motion of the end-effector achieved by the ISSIOF method. Similar phenomena are observed in Fig. 18(c) and (d). Additionally, at the lower feedrate command of 5 mm/s, both methods show better surface quality. These observations demonstrate the effectiveness of the proposed ISSIOF method in reducing surface roughness.

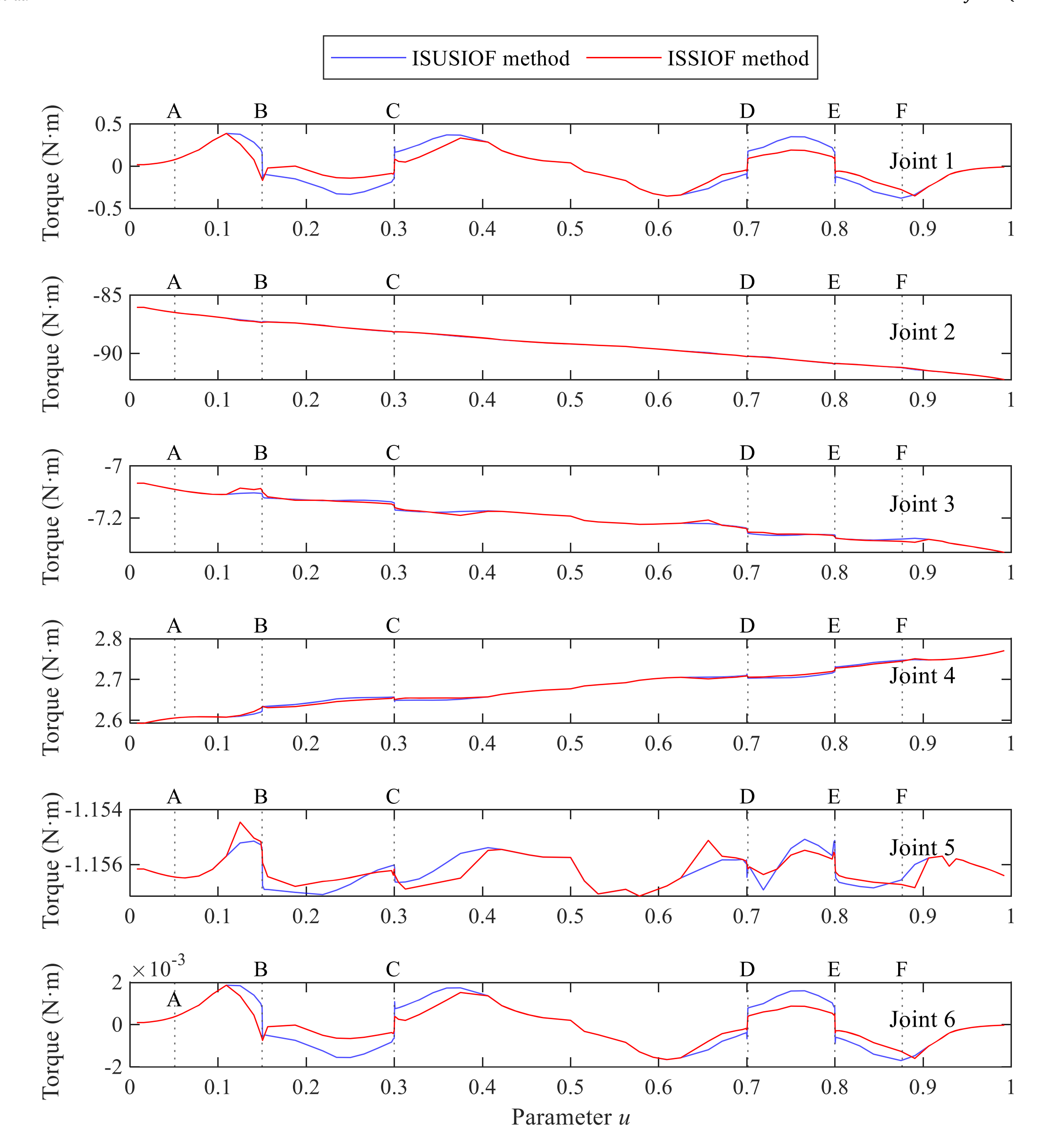


Fig. 13. Joint torques before and after considering the torque change rate constraint when $F = 10$ mm/s.

Table 2
Comparison results of processing time.

	ISUSIOF (s)	ISSIOF (s)	Increase (%)
$F = 5$ mm/s	25.9090	26.1850	1.0653
$F = 10$ mm/s	12.9450	12.9930	0.3708

Table 3
Comparison results of path velocity fluctuation.

	ISUSIOF (mm/s)	ISSIOF (mm/s)	Decrease (%)
$F = 5$ mm/s	0.0098	0.0010	89.7959
$F = 10$ mm/s	0.0151	0.0020	86.7550

Table 4
Comparison results of acceleration fluctuation.

	ISUSIOF (mm/s^2)	ISSIOF (mm/s^2)	Decrease (%)
$F = 5$ mm/s	1.5619	0.1029	93.4119
$F = 10$ mm/s	7.1926	0.3330	95.3702

Table 5
Comparison results of jerk fluctuation.

	ISUSIOF (mm/s^3)	ISSIOF (mm/s^3)	Decrease (%)
$F = 5$ mm/s	283.5369	69.5136	75.4834
$F = 10$ mm/s	1683.1359	208.3961	87.6186

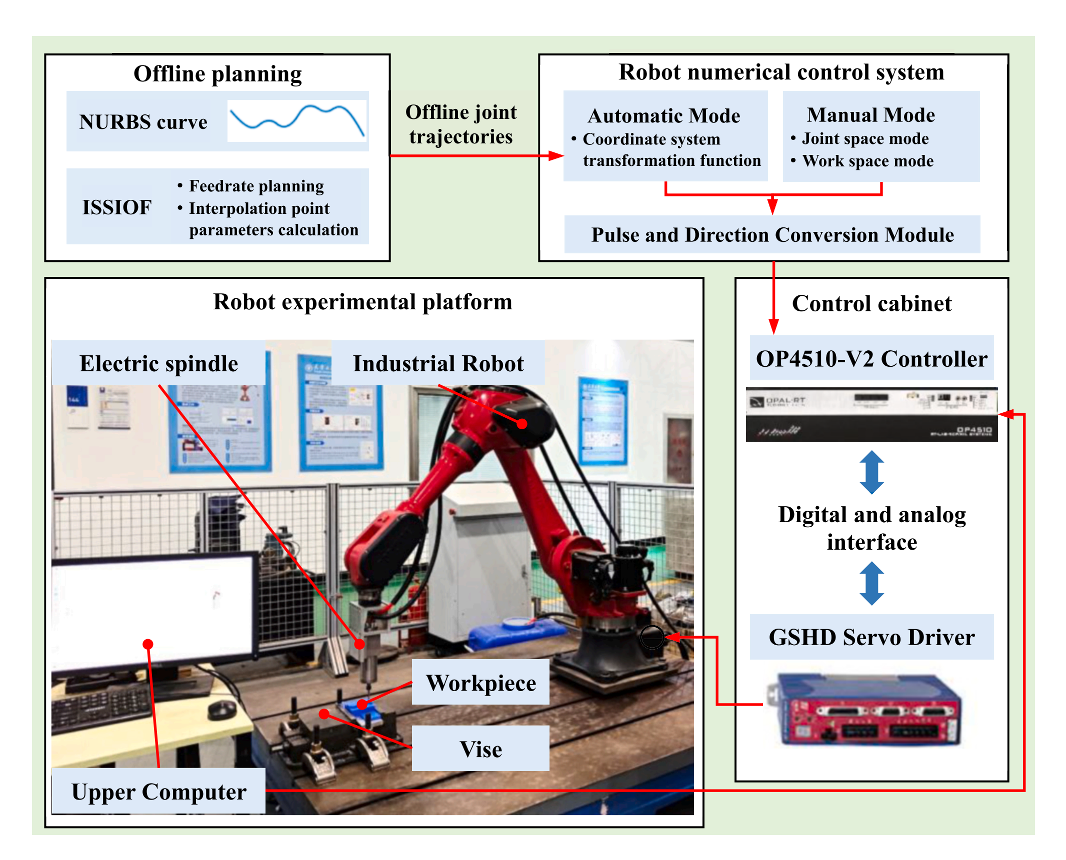


Fig. 14. The schematic diagram of the robot experimental platform construction.

In Fig. 20(a), the blue curve is significantly higher than the red curve before region A. Similarly, in Fig. 20(b), the blue curve is noticeably higher than the red curve before region B. This is due to the presence of acceleration, constant feedrate, and deceleration processes in this area. The proposed ISSIOF method reduces vibrations in this region by minimizing the maximum acceleration and smoothing the interpolation output feedrate and acceleration. Between C-to-D in Fig. 20(b), the blue curve is slightly higher than the red curve. This is because this range primarily consists of a constant feedrate region. The proposed ISSIOF method effectively reduces velocity fluctuations, leading to decreased vibrations in this constant feedrate region. In the D-to-E regions of both (a) and (b), the blue curve is higher than the red curve. This is mainly because the ISSIOF method smooths the acceleration in this area, effectively reducing vibrations. After region F in Fig. 20(b), the blue curve is again higher than the red curve. This is primarily because the ISSIOF method minimizes the maximum jerk in this region, effectively reducing vibrations.

As shown in Fig. 21, the primary profile obtained by the proposed ISSIOF method is closer to the ideal profile. The specific contour errors of both methods are shown in Fig. 22. In most regions of Fig. 22(a) and (b), the blue curve is higher than the red curve. Before region A, both curves change from high to low, and the height difference is not significant. This behavior can be attributed to the initial cutting stage, where the cutting width gradually transitions from zero to the specified width. After region A, both curves begin to rise, reaching a minimum at region C. The reason for the analysis is that region A in Fig. 22 corresponds to a transition from a lower

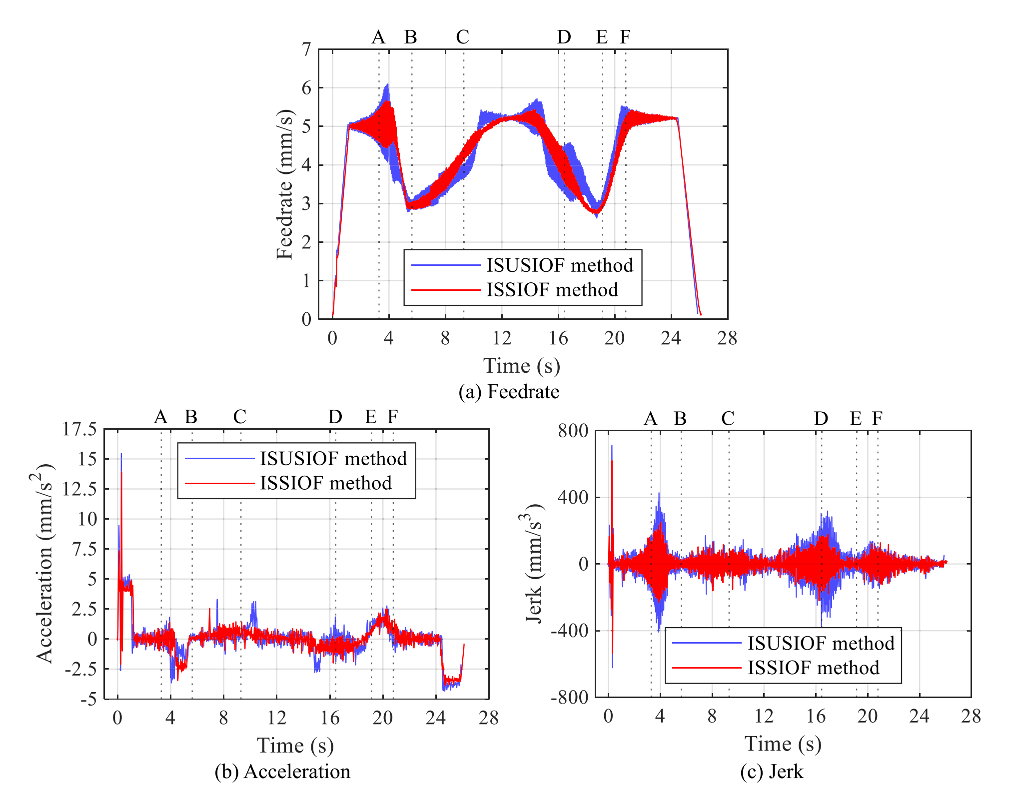


Fig. 15. Experimental results of robot end-effector when $F = 5$ mm/s.
(a) Feedrate, (b) Acceleration, (c) Jerk.

curvature area to a higher curvature area, followed by a decrease in curvature towards region C, where the milling process becomes more stable. Between regions C and D, the blue curve is significantly higher than the red curve. This is because this range primarily consists of a constant feedrate region. The proposed ISSIOF method reduces velocity fluctuations effectively, resulting in decreased vibrations in the constant feedrate region. In the region from E to F, both curves change from low to high and then back to low. The analysis indicates that region E in Fig. 22 corresponds to a transition point where the curvature switches from high to low and then back to high. At region F, the curvature starts decreasing again, leading to a more stable milling process.

From Table 6, the results show that ISSIOF can significantly reduce roughness Ra compared to ISUIOF, particularly at a feedrate command of 10 mm/s. This indicates that smooth interpolation output feedrate has a significant effect on reducing roughness Ra.

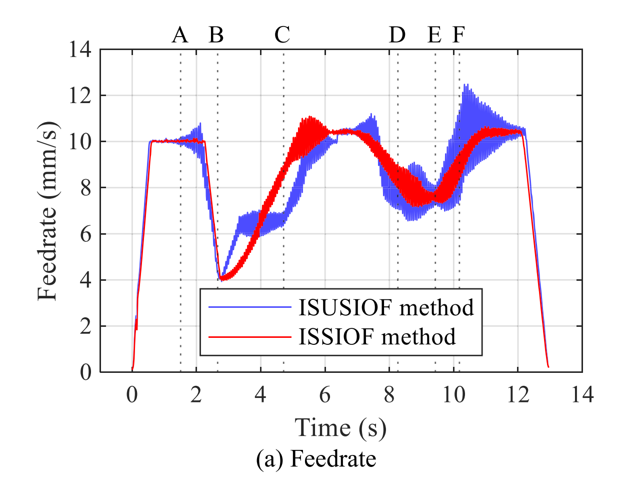
As shown in Tables 7 and 8, both methods increase the average and maximum contour errors at a feedrate command of 10 mm/s compared to 5 mm/s. Furthermore, ISSIOF demonstrates a significant reduction in average and maximum contour errors compared to ISUIOF, with the improvement effect more pronounced at 10 mm/s. This indicates that the overall contour errors are larger at higher feedrate commands. This is because of the relatively low rigidity of the robot employed in this study. According to the literature [46–48], dynamic effects become more prominent at higher feedrate commands, leading to increased contour errors. Smooth interpolation output feedrate has a certain degree of improvement in contour accuracy. The proposed ISSIOF method effectively smooths the interpolation output feedrate, resulting in a more significant reduction in contour error at higher feedrate commands.

6. Conclusion

In the paper, a NURBS curve interpolation strategy for smooth interpolation output feedrate is proposed. The strategy smooths the input commands of the robot and thus reduces contour error and roughness. The conclusions are as follows:

(1) The simulation results show that ISSIOF can significantly smooth the interpolation output feedrate, ensuring smooth motion of the robot’s end-effector with little efficiency loss.
(2) Flank milling experiments demonstrate that ISSIOF can reduce contour error and significantly reduce roughness.
(3) The simulation and experimental results show that smooth interpolation output feedrate can effectively reduce roughness and contour errors.
(4) The proposed ISSIOF significantly reduces roughness and contour errors at higher feedrate commands.

The limitations of the proposed strategy and the corresponding potential directions for future research mainly involve the following



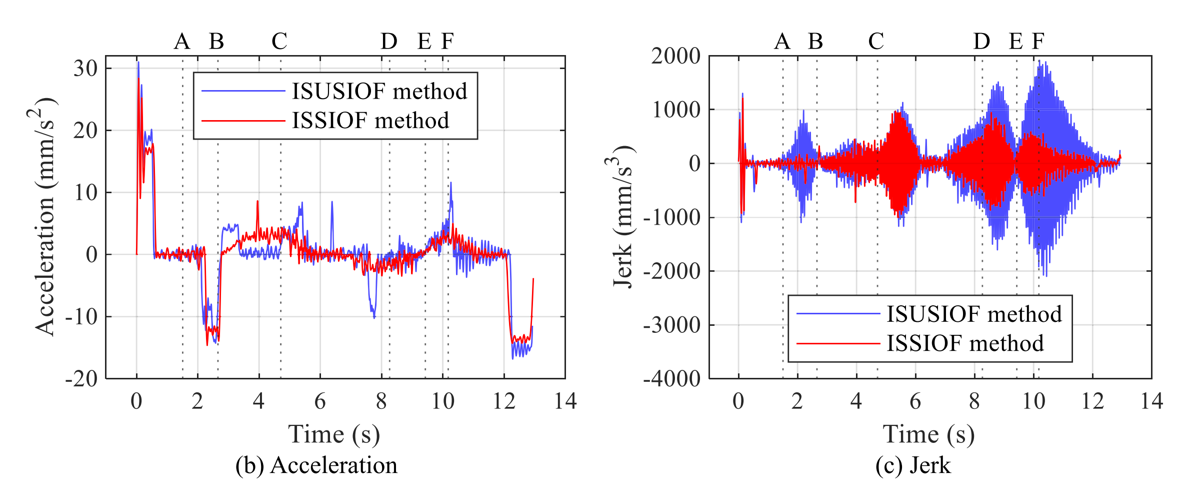


Fig. 16. Experimental results of robot end-effector when $F = 10$ mm/s.
(a) Feedrate, (b) Acceleration, (c) Jerk.

two aspects. First, the current strategy is only applicable to single-segment NURBS curve, whereas in practical industrial applications, part machining often involves multi-segment paths consisting of both NURBS curves and short line segments. Second, the planned feedrate for flank milling machining is relatively low. Future research will focus on expanding the method's applicability to multi-segment paths combining NURBS curves and short line segments, as well as incorporating cutting forces and other process factors to increase the feedrate without sacrificing accuracy.

CRediT authorship contribution statement

Yonghao Guo: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Wentie Niu:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Hongda Liu:** Writing – review & editing, Investigation. **Zengao Zhang:** Writing – review & editing, Investigation. **Hao Zheng:** Writing – review & editing, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

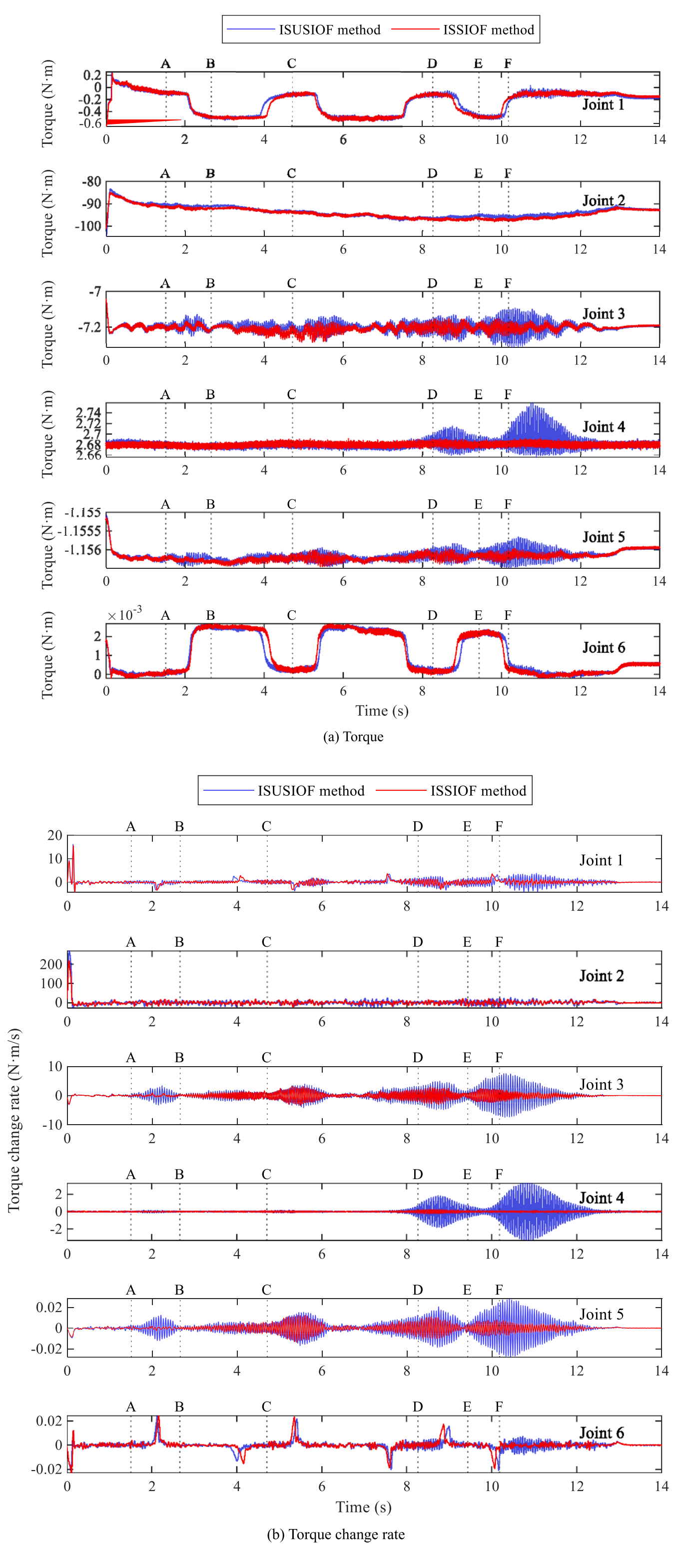


Fig. 17. Joint torques and torque change rate when $F = 10$ mm/s.
(a) Torque, (b) Torque change rate.

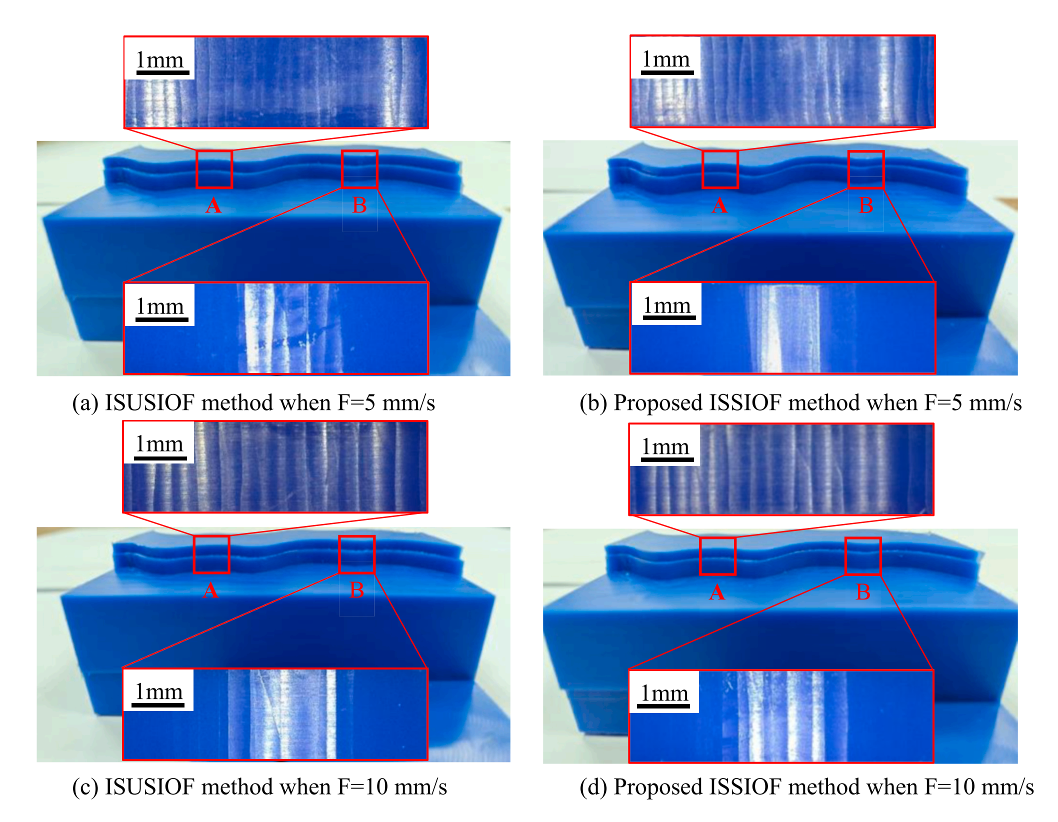


Fig. 18. Flank milling experiments results.
(a) ISUSIOF method when $F = 5$ mm/s, (b) Proposed ISSIOF method when $F = 5$ mm/s, (c) ISUSIOF method when $F = 10$ mm/s, (d) Proposed ISSIOF method when $F = 10$ mm/s.

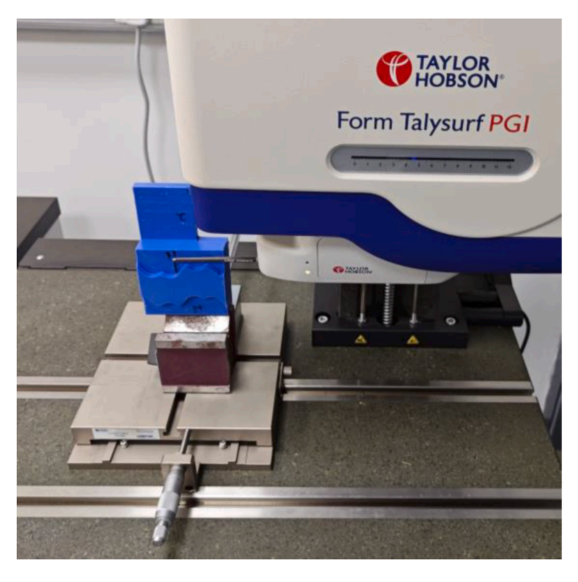


Fig. 19. Schematic diagram of flank milling profile measurement.

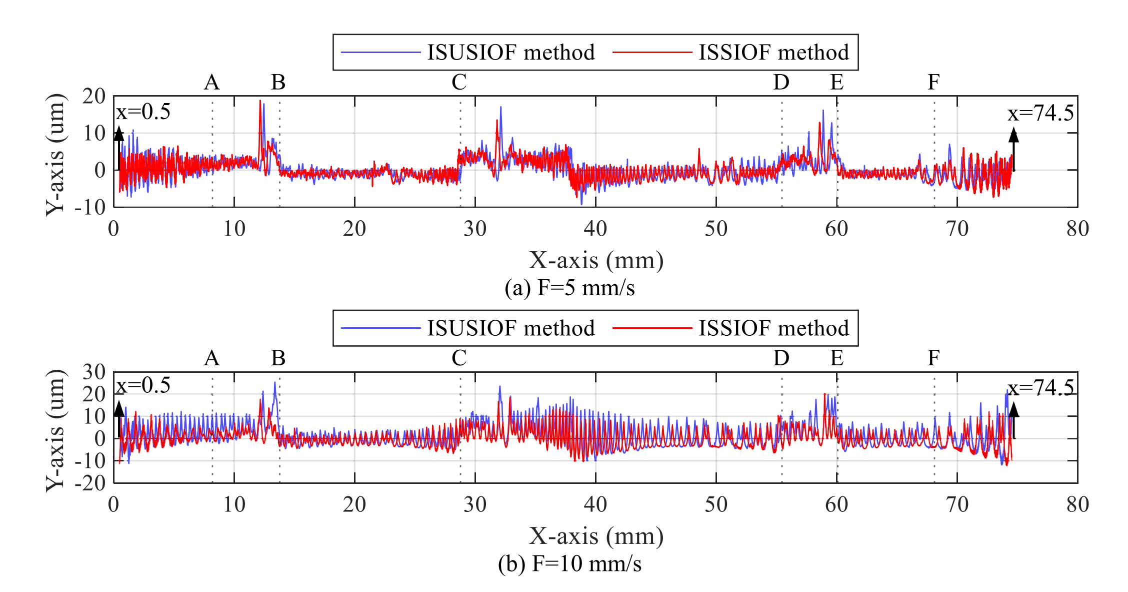


Fig. 20. Experimental results of the roughness profile.
(a) $F = 5$ mm/s, (b) $F = 10$ mm/s.

Table 6
Comparison results of Ra.

	ISUSIOF (um)	ISSIOF (um)	Decrease (%)
$F = 5$ mm/s	2.0247	1.9446	3.9561
$F = 10$ mm/s	3.9342	3.0117	23.4482

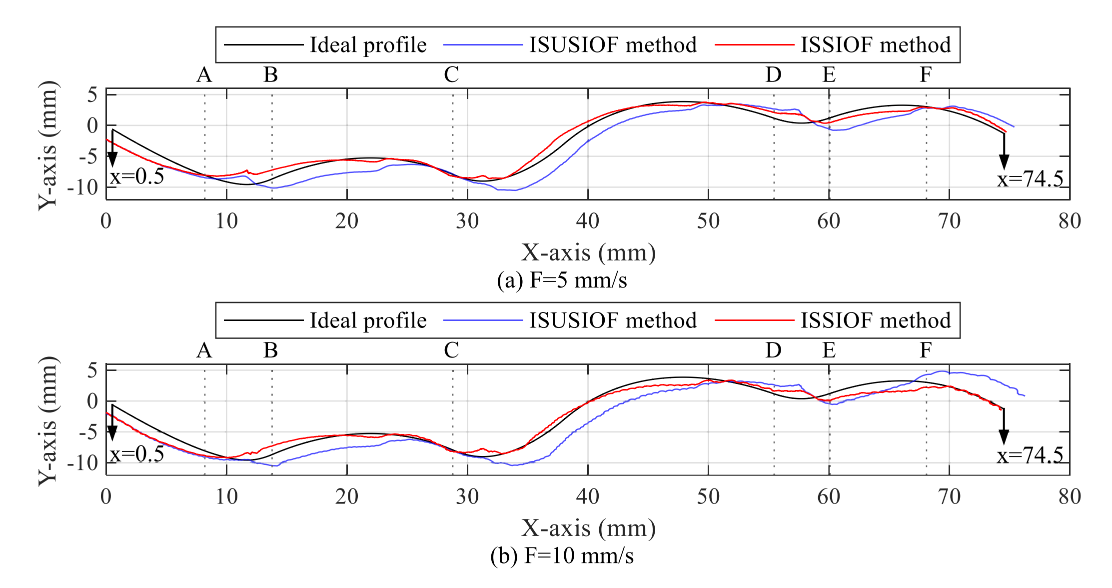


Fig. 21. Experimental results of the primary profile (contour error magnified 8x).
(a) $F = 5$ mm/s, (b) $F = 10$ mm/s.

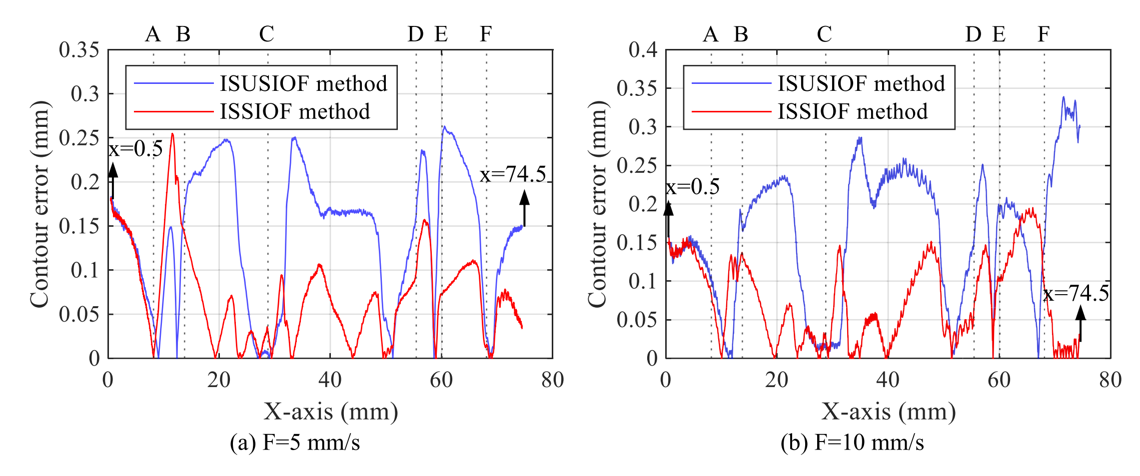


Fig. 22. Experimental results of contour error.
(a) $F = 5$ mm/s, (b) $F = 10$ mm/s.

Table 7
Comparison results of average contour error.

	ISUSIOF (mm)	ISSIOF (mm)	Decrease (%)
$F = 5$ mm/s	0.1414	0.0704	50.2122
$F = 10$ mm/s	0.1622	0.0754	53.5142

Table 8
Comparison results of max contour error.

	ISUSIOF (mm)	ISSIOF (mm)	Decrease (%)
$F = 5$ mm/s	0.2633	0.2550	3.1523
$F = 10$ mm/s	0.3393	0.1953	42.4403

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability

Data will be made available on request.

References

[1] Y. Fang, J. Hu, W.H. Liu, Q.Q. Shao, J. Qi, Y.H. Peng, Smooth and time-optimal S-curve trajectory planning for automated robots and machines, Mech. Mach. Theory 137 (2019) 127–153. https://doi.org/10.1016/j.mechmachtheory.2019.03.019.
[2] B. Siciliano, O. Khatib, Springer Handbook of Robotics, Springer, Cham, 2016, https://doi.org/10.1007/978-3-319-32552-1 second ed.
[3] M. Liu, Y. Huang, L. Yin, J.W. Guo, X.Y. Shao, G.J. Zhang, Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools, Int. J. Mach. Tools Manuf. 87 (2014) 1–15, https://doi.org/10.1016/j.ijmachtools.2014.07.002.
[4] G.X. Wang, Q.L. Shu, J. Wang, L. Li, Research on adaptive non-uniform rational B-spline real-time interpolation technology based on acceleration constraints, Int. J. Adv. Manuf. Technol. 91 (2017) 2089–2100, https://doi.org/10.1007/s00170-016-9914-4.
[5] S.J. Ji, L.G. Lei, J. Zhao, X.Q. Lu, H. Gao, An adaptive real-time NURBS curve interpolation for 4-axis polishing machine tool, robot. Comput.-Integr. Manuf. 67 (2021) 102025, https://doi.org/10.1016/j.rcim.2020.102025.
[6] J. Zhou, H. Yi, H.J. Cao, P. Jiang, C.Y. Zhang, W.W. Ge, Structural decomposition-based energy consumption modeling of robot laser processing systems and energy-efficient analysis, robot. Comput.-Integr. Manuf. 76 (2022) 102327, https://doi.org/10.1016/j.rcim.2022.102327.
[7] H.J. Cao, J. Zhou, P. Jiang, K.K.B. Hon, H. Yi, C.Y. Dong, An integrated processing energy modeling and optimization of automated robotic polishing system, robot. Comput.-Integr. Manuf. 65 (2020) 101973, https://doi.org/10.1016/j.rcim.2020.101973.
[8] J.W. Ma, S. Gao, H.T. Yan, Q. Lv, G.Q. Hu, A new approach to time-optimal trajectory planning with torque and jerk limits for robot, robot. Auton. Syst. 140 (2021) 103744, https://doi.org/10.1016/j.robot.2021.103744.
[9] A. Nagy, I. Vajk, Sequential time-optimal path-tracking algorithm for robots, IEEE Trans. Robot. 35 (2019) 1253–1259, https://doi.org/10.1109/Tro.2019.2920090.
[10] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, M. Diehl, Time-optimal path tracking for robots: a convex optimization approach, IEEE Trans. Autom. Control 54 (2009) 2318–2327, https://doi.org/10.1109/Tac.2009.2028959.
[11] L. Lu, J. Zhang, J.Y.H. Fuh, J. Han, H. Wang, Time-optimal tool motion planning with tool-tip kinematic constraints for robotic machining of sculptured surfaces, Robot. Comput.-Integr. Manuf. 65 (2020) 101969, https://doi.org/10.1016/j.rcim.2020.101969.

[12] S.D. Butler, M. Moll, L.E. Kavraki, A general algorithm for time-optimal trajectory generation subject to minimum and maximum constraints, in: K. Goldberg, A. Pieter, B. Kostas, M. Lauren (Eds.), Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, Springer International Publishing, Cham, 2020, pp. 368–383, https://doi.org/10.1007/978-3-030-43089-4_24.
[13] K.M. Lynch, F.C. Park, Modern Robotics: Mechanics, Planning, and Control, Cambridge University Press, Cambridge, UK, 2017.
[14] Q.C. Pham, A. General, Fast, and robust implementation of the time-optimal path parameterization algorithm, IEEE Trans. Robot. 30 (2014) 1533–1540, https://doi.org/10.1109/Tro.2014.2351113.
[15] K. Shin, N. McKay, Minimum-time control of robotic manipulators with geometric path constraints, IEEE Trans. Autom. Control 30 (1985) 531–541, https://doi.org/10.1109/TAC.1985.1104009.
[16] J.E. Bobrow, S. Dubowsky, J.S. Gibson, Time-optimal control of robotic manipulators along specified paths, Int. J. Robot. Res. 4 (1985) 3–17, https://doi.org/10.1177/027836498500400301.
[17] A. Nagy, I. Vajk, LP-based velocity profile generation for robotic manipulators, Int. J. Control 91 (2018) 582–592, https://doi.org/10.1080/00207179.2017.1286535.
[18] K. Hauser, Fast interpolation and time-optimization with contact, Int. J. Robot. Res. 33 (2014) 1231–1250, https://doi.org/10.1177/0278364914527855.
[19] J.T. Betts, W.P. Huffman, Path-constrained trajectory optimization using sparse sequential quadratic programming, J. Guid. Control Dyn. 16 (1993) 59–68, https://doi.org/10.2514/3.11428.
[20] Y.H. Guo, W.T. Niu, J.P. Zhou, H.D. Liu, Near-time optimal feedrate planning for the NURBS curve considering interpolation error constraints, robot. Comput.-Integr. Manuf. 86 (2024), https://doi.org/10.1016/j.rcim.2023.102679.
[21] D. Kaserer, H. Gattringer, A. Muller, Nearly optimal path following with jerk and torque rate limits using dynamic programming, IEEE Trans. Robot. 35 (2019) 521–528, https://doi.org/10.1109/Tro.2018.2880120.
[22] M. Oberherber, H. Gattringer, A. Muller, Successive dynamic programming and subsequent spline optimization for smooth time optimal robot path tracking, Mech. Sci. 6 (2015) 245–254, https://doi.org/10.5194/ms-6-245-2015.
[23] S. Singh, M.C. Leu, Optimal trajectory generation for robotic manipulators using dynamic programming, J. Dyn. Syst., Meas., Control 109 (1987) 88–96, https://doi.org/10.1115/1.3143842.
[24] K. Shin, N. McKay, A dynamic programming approach to trajectory planning of robotic manipulators, IEEE Trans. Autom. Control 31 (1986) 491–500, https://doi.org/10.1109/TAC.1986.1104317.
[25] Y. Fang, J. Qi, J. Hu, W.M. Wang, Y.H. Peng, An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints, Mech. Mach. Theory 153 (2020) 103957, https://doi.org/10.1016/j.mechmachtheory.2020.103957.
[26] H. Wang, H. Wang, J.H. Huang, B. Zhao, L. Quan, Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve, Mech. Mach. Theory 139 (2019) 284–293, https://doi.org/10.1016/j.mechmachtheory.2019.05.002.
[27] Y.H. Li, T. Huang, D.G. Chetwynd, An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines, Mech. Mach. Theory 126 (2018) 479–490, https://doi.org/10.1016/j.mechmachtheory.2018.04.026.
[28] Y.F. Hu, X. Jiang, G.Y. Huo, C. Su, S.W. Zhou, B.L. Wang, H.X. Li, Z.M. Zheng, A novel feed rate scheduling method with acc-jerk-continuity and round-off error elimination for non-uniform rational B-spline interpolation, J. Comput. Des. Eng. 10 (2023) 294–317, https://doi.org/10.1093/jcde/qwad004.
[29] H.P. Ni, C.R. Zhang, Q.Z. Chen, S. Ji, T.L. Hu, Y.N. Liu, A novel time-rounding-up-based feedrate scheduling method based on S-shaped ACC/DEC algorithm, Int. J. Adv. Manuf. Technol. 104 (2019) 2073–2088, https://doi.org/10.1007/s00170-019-03882-0.
[30] H.P. Ni, T.L. Hu, C.R. Zhang, S. Ji, Q.Z. Chen, An optimized feedrate scheduling method for CNC machining with round-off error compensation, Int. J. Adv. Manuf. Technol. 97 (2018) 2369–2381, https://doi.org/10.1007/s00170-018-1986-x.
[31] Q. Liu, H. Liu, S.M. Yuan, High accurate interpolation of NURBS tool path for CNC machine tools, Chin. J. Mech. Eng. 29 (2016) 911–920, https://doi.org/10.3901/Cjme.2016.0407.047.
[32] S. Ji, T.L. Hu, Z.G. Huang, C.R. Zhang, A NURBS curve interpolator with small feedrate fluctuation based on arc length prediction and correction, Int. J. Adv. Manuf. Technol. 111 (2020) 2095–2104, https://doi.org/10.1007/s00170-020-06258-x.
[33] K. Zhao, S.R. Li, Z.J. Kang, Smooth minimum time trajectory planning with minimal feed fluctuation, Int. J. Adv. Manuf. Technol. 105 (2019) 1099–1111, https://doi.org/10.1007/s00170-019-04308-7.
[34] X.F. Li, H. Zhao, X.M. He, H. Ding, A novel cartesian trajectory planning method by using triple NURBS curves for industrial robots, Robot. Comput.-Integr. Manuf. 83 (2023), https://doi.org/10.1016/j.rcim.2023.102576.
[35] Z.Y. Jia, D.N. Song, J.W. Ma, G.Q. Hu, W.W. Su, A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints, Int. J. Mach. Tools Manuf. 116 (2017) 1–17, https://doi.org/10.1016/j.ijmachtools.2016.12.007.
[36] H. Zhao, L.M. Zhu, H. Ding, A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction, Int. J. Mach. Tools Manuf. 75 (2013) 1–8, https://doi.org/10.1016/j.ijmachtools.2013.08.002.
[37] X.T. Zhang, Z. Song, An iterative feedrate optimization method for real-time NURBS interpolator, Int. J. Adv. Manuf. Technol. 62 (2012) 1273–1280, https://doi.org/10.1007/s00170-011-3847-8.
[38] M.C. Tsai, C.W. Cheng, A real-time predictor-corrector interpolator for CNC machining, J. Manuf. Sci. Eng. 125 (2003) 449–460, https://doi.org/10.1115/1.1578670.
[39] M.Y. Cheng, M.C. Tsai, J.C. Kuo, Real-time NURBS command generators for CNC servo controllers, Int. J. Mach. Tools Manuf. 42 (2002) 801–813, https://doi.org/10.1016/S0890-6955(02)00015-9.
[40] W.T. Lei, M.P. Sung, L.Y. Lin, J.J. Huang, Fast real-time NURBS path interpolation for CNC machine tools, Int. J. Mach. Tools Manuf. 47 (2007) 1530–1541, https://doi.org/10.1016/j.ijmachtools.2006.11.011.
[41] T. Zhang, L.L. Guo, Y.B. Zou, A NURBS Curve Interpolator Based on Double-Step Signal and Finite Impulse Response Filters, J. Manuf. Sci. Eng. 145 (2023) 021001, https://doi.org/10.1115/1.4055246.
[42] J.C. Wu, H.C. Zhou, X.Q. Tang, J.H. Chen, A NURBS interpolation algorithm with continuous feedrate, Int. J. Adv. Manuf. Technol. 59 (2012) 623–632, https://doi.org/10.1007/s00170-011-3520-2.
[43] K. Erkorkmaz, Y. Altintas, Quintic spline interpolation with minimal feed fluctuation, J. Manuf. Sci. Eng. 127 (2005) 339–349, https://doi.org/10.1115/1.1830493.
[44] M. Heng, K. Erkorkmaz, Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability, Int. J. Mach. Tools Manuf. 50 (2010) 281–293, https://doi.org/10.1016/j.ijmachtools.2009.11.005.
[45] J. Wu, H. Ye, G. Yu, T. Huang, A novel dynamic evaluation method and its application to a 4-DOF parallel manipulator, Mech. Mach. Theory 168 (2022) 104627, https://doi.org/10.1016/j.mechmachtheory.2021.104627.
[46] J. Wu, Y. Gao, B.B. Zhang, L.P. Wang, Workspace and dynamic performance evaluation of the parallel manipulators in a spray-painting equipment, robot. Comput.-Integr. Manuf. 44 (2017) 199–207, https://doi.org/10.1016/j.rcim.2016.09.002.
[47] J. Wu, G. Yu, Y. Gao, L.P. Wang, Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool, Mech. Mach. Theory 121 (2018) 430–445, https://doi.org/10.1016/j.mechmachtheory.2017.10.023.
[48] J. Wu, J.S. Wang, L.P. Wang, T.M. Li, Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy, Mech. Mach. Theory 44 (2009) 835–849, https://doi.org/10.1016/j.mechmachtheory.2008.04.002.